Università
Ca'Foscari
Venezia

Scuola Dottorale di Ateneo
Graduate School

Dottorato di ricerca in Informatica
Ciclo 28
Anno di discussione 2015

Algebraic structures for the lambda calculus and the propositional logic

SETTORE SCIENTIFICO DISCIPLINARE DI AFFERENZA: INF/01

Tesi di Dottorato di Giordano Favro, matricola 955977

Coordinatore del Dottorato

Prof. Riccardo Focardi

Tutori del Dottorando

Prof. Antonio Bucciarelli

Prof. Antonino Salibra

Università Ca' Foscari Venezia
Dottorato di Ricerca in Informatica, XXVIII ciclo

Ph.D. Thesis

Algebraic structures for the lambda calculus and the propositional logic

Giordano Favro

Supervisor
Antonino Salibra

Supervisor
Antonio Bucciarelli

PhD Coordinator
Riccardo Focardi

PhD Coordinator
Ahmed Bouajjani

Referee
Ugo de'Liguoro

Referee
Francesco Paoli

February, 2016

Author’s Web Page: http://www.unive.it/persone/giordano.favro

Author’s e-mail: giordano.favro@unive.it

Author’s address:

Dipartimento di Informatica
Università Ca’ Foscari di Venezia
Via Torino, 155
30172 Venezia Mestre – Italia
tel. +39 041 2348411
fax. +39 041 2348419
web: `http://www.dsi.unive.it`

A Maria Grazia

Abstract

Part I

Among the unsolvable terms of the lambda calculus, the mute ones are those having the highest degree of undefinedness. For each natural number $n \geq 1$, we introduce two infinite and recursive sets $\mathcal{M}_n$ and $\mathcal{G}_n$. Their elements are called *restricted regular mute* and *regular mute* terms respectively. They are defined inductively and we prove that they are mute.

Furthermore, we show that sets $\mathcal{M}_n$ are *graph-easy* for any n: for any closed term t there exists a graph model equating all the terms of $\mathcal{M}_n$ to t. We also provide a brief survey of the notion of undefinedness in λ-calculus.

Part II

We introduce *factor algebras of first-order type* and show that they can be used to provide an algebraic counterpart of ordinary first-order structures. We show that this translation can be extended to open formulas and equations between terms. By considering that propositional logic is a first-order logic on a particular type τ_{CL}, a new algebraic calculus for propositional logic is developed. Rules for the calculus are the axioms of the variety generated by the factor algebras of type τ_{CL}. We also provide a confluent and terminating term rewriting system for such calculus.

Furthermore, we study the basic algebraic properties of factor algebras of first-order type through the notion of splitting pair.

Resumé

Partie I

Parmi les termes non résolubles du lambda-calcul, les termes muets sont ceux dont le “degré d’indéfini” est maximum. Pour chaque nombre naturel $n \geq 1$, nous introduisons deux ensembles infinis et récursifs de lambda-termes, $\mathcal{M}_n$ et $\mathcal{G}_n$. Nous appelons leurs éléments “termes muets réguliers restreints” et “termes muets réguliers”, respectivement, et nous prouvons qu’il s’agit bien de termes muets.

Nous prouvons ensuite que les ensembles $\mathcal{M}_n$ sont “graph easy”: pour chaque terme clos t du lambda-calcul, il existe un modéle de graphe qui égalise t et tout les éléments de $\mathcal{M}_n$.

Partie II

Nous introduisons les “factor algebras of first-order type”, qui peuvent être utilisés pour algébriser la notion de structure du premier ordre et de formule ouverte. Nous analysons les propriétés algébriques de base des “factor algebras of first order type” en utilisant la notion de “splitting pair”.

En nous appuyant sur le fait que la logique propositionnelle est une logique du premier ordre sur un type particulier τ_{CL}, nous développons un nouveau calcul algébrique pour la logique propositionnelle: ses régles sont les axiomes de la variété des “factor algebras” du type τ_{CL}. Nous présentons un sistéme de réécriture confluent et terminant pour ce calcul.

Sommario

Parte I

Nel lambda calcolo i termini unsolvable sono quelli che mostrano il più alto livello di indefinitezza. Per ogni naturale $n \geq 1$, introduciamo due insiemi infiniti e ricorsivi, $\mathcal{M}_n$ e $\mathcal{G}_n$. I loro elementi sono definiti induttivamente e sono chiamati rispettivamente *restricted regular mute terms* e *regular mute terms*: di questi termini mostriamo che appartengono alla classe dei termini muti.

Proviamo inoltre che gli insiemi $\mathcal{M}_n$ sono *graph easy*: per ogni termine chiuso t dimostriamo che esiste un graph model in cui l'interpretazione di ogni elemento di $\mathcal{M}_n$ è uguale all'interpretazione di t. Forniamo inoltre una breve dissertazione sulla nozione di indefinitezza nel lambda calcolo.

Parte II

Nella seconda parte introduciamo le *factor algebras of first-order type* e mostriamo come esse possano essere usate per fornire una controparte algebrica per le strutture del primo ordine. Mostriamo inoltre che questa traduzione può essere estesa alle formule aperte e alle equazioni fra termini.

Considerando che la logica proposizionale è una logica del primo ordine su un particolare tipo τ_{CL}, sviluppiamo un nuovo calcolo algebrico per la logica proposizionale, le cui regole sono date dagli assiomi che generano la varietà delle factor algebra di tipo τ_{CL}. Di questo calcolo forniamo inoltre un sistema di riscrittura che si prova essere confluente e terminante.

Studiamo inoltre le proprietà algebriche di base delle factor algebras of first-order type tramite la nozione di unsplitting pair.

Acknowledgments

I would like to express my gratitude to my advisors Antonino Salibra and Antonio Bucciarelli for their guidance, help and patience during these three years. Thanks also for all that they thought me.
Many thanks also to Alberto Carraro and Giulio Manzonetto for the interesting, and sometimes very challenging, work we had together.
Thanks to Ugo de'Liguoro and to Francesco Paoli for accepting to review my Ph.D. thesis.
Thanks to Thomas Ehrhard, Alexis Saurin, Damiano Mazza and Claudia Faggian for their kind aid while attending courses at Paris 7.
I would like to thanks all the persons with whom I had good time in Mestre and Paris: in particular, thanks to Enrico, Marco and Teresa; thanks to the PPS Ph.D. students and to Claudio. Thanks to Cécile and to Lara for their help while staying in Paris.

A very special thanks goes to Giulia.
Last but not least, thanks to my mother Franca and to my father Gianni for their support.

Preface

This thesis consists of two parts. The first one deals with the Lambda Calculus, while the second one is about Universal Algebra and Algebraic Logic. The parts are independent, so each has its own introduction and preliminaries.

The first part is based on a paper accepted for publication under minor revision in the journal Theoretical Computer Science ([21]). Its results were proved in joint work with my supervisors Antonino Salibra and Antonio Bucciarelli and with Alberto Carraro. Here are presented some extensions of the original results.

In second part we present some unpublished results on universal algebra and algebraic logic. Its contents are fruits of a joint work with Antonino Salibra. The proof of confluence and termination of the rewriting system are mainly obtained by Giulio Manzonetto.

Contents

I

Regular mute terms

1

Introduction

The problem of characterizing λ-terms which represent an undefined computational process is an important issue, that have been analyzed since the beginning of λ-calculus.

In order to show that λ-calculus is Turing-complete, in fact, it is necessary to introduce a sensible notion of *undefined λ-term*. Kleene ([46]) showed in 1936 that the Herbrand-Gödel's general recursive functions are λ-representable, but somehow he eluded the general problem. He considered only total recursive functions, so, for example, it is not clear what is the meaning of the application of Kleene's λ-term implementing the μ operator to a λ-term defining a unary function that never returns 0.

In early '40s, Church ([27]) extended Kleene's result. He showed that every recursive function f can be λ-represented by considering terms having no normal form as the undefined values of f. This means that it is possible to find a λ-term t in such a way that, if $f(n)$ is not defined, then $t\lceil n\rceil$ does not have a normal form ($\lceil n\rceil$ denotes the Church numeral for n).

It follows that the set of "undefined" λ-terms must be considered as a subset of the non-normalizing ones. The identification "undefined=without normal form" can be accepted for λI-calculus, but strong syntactical and semantical considerations ([6] pp- 38-43) prevent us from accepting it in the λK-calculus.

A better characterization is provided by *unsolvable λ-terms*. Unsolvable terms are defined operationally, as the terms that never exhibit a specific stable form along reductions, i.e., they are never in head normal form. Unsolvables show strong properties of undefinedness ([6], pp. 42-43). Furthermore, some important non-normalizing terms like Turing fixed-point operator are not considered as meaningless anymore.

Given a partial function $f : \omega^p \to \omega$ and a set of closed terms $A \subseteq \Lambda^0$, we say that f is *λ-representable with A as set of undefined elements* if there exists a λ-term F such that, for any p-uple of natural numbers $n_1, \dots, n_p$,

$$F\lceil n_1\rceil \dots \lceil n_p\rceil = \begin{cases} \lceil f(n_1, \dots, n_p)\rceil & \text{if } (n_1, \dots, n_p) \text{ belongs to } Dom(f); \\ \text{a term belonging to } A & \text{otherwise.} \end{cases}$$

The following theorem of Statman [62] (Theorem 3.3.6) provides a sufficient condition for a set of terms to characterize the undefined values of a partial function.

> Every partial function is λ-representable with respect to any nonempty β-closed set of closed λ-terms that is the complement of a recursively enumerable set.

In particular, the sets of unsolvables ([5]), easy ([68]) and zero terms ([61]) suitably represent undefinedness. A set M of λ-terms is an *easy set* if for any closed term t, the λ-theory $\{t = s : s \in M\}$ is consistent. A term t is *easy* if $\{t\}$ is an easy set. Zero terms are unsolvables that cannot be reduced to a term of the form $\lambda x.t$.

Easiness is an important notion that provides powerful tools for studying the models of the λ-calculus. Jacopini [39] syntactically proved that the paradigmatic unsolvable term $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$ is easy. Baeten and Boerboom [4] gave the first semantical proof of this result by showing that for any closed term t one can build a graph model satisfying the equation $\Omega = t$. This semantical result extends to other classes of models and to some other terms which have a behavior similar to Ω (cf. [11, 12] for a survey of such results).

Terms Ω and ΩI are easy terms, but they cannot be consistently equated to K. This implies that $\{\Omega, \Omega I\}$ is not an easy set, and *a fortiori*, that the set of easy terms is not an easy set.

Mute λ-terms have been introduced by Berarducci [8]. Mute terms are defined operationally, as 0-terms which are not β-convertible to a 0-term applied to something else. Berarducci proved that the set of mute terms is an easy set: this implies that somehow they are "more undefined" than easy terms. He also built a non-sensible model of λ-calculus in which all mute terms are identified.

Given a class of models $\mathcal{C}$, it is possible to analyze easiness of terms and sets with respect to $\mathcal{C}$. More formally: given a class $\mathcal{C}$ of models of λ-calculus, and an easy set S, we say that S is $\mathcal{C}$-easy if, for every closed term t, there exists a model in $\mathcal{C}$ which equates all the terms in S to t.

$\mathcal{C}$-easiness gives insights on the expressive power of the class $\mathcal{C}$. For instance, it had been conjectured ([3]) that any easy term t was filter-easy, i.e., that t is easy w.r.t. the class of filter models. Carraro and Salibra [25] showed that this is not the case.

Graph models are arguably the simplest models of the λ-calculus, since they are described by a denumerable set and a injective function. The most flexible method for building graph models is *forcing*. It was introduced by Baeten and Boerboom ([4]) and it consists in completing a partial model into a total one. The forcing method depends not only on the initial partial model but also on the consistency problem one is interested in. The method was afterwards generalized to other classes of webbed models by Jiang [41] and Kerth [45]. It was also generalized to families of terms similar to Ω by Zylberajch [69] and Berline–Salibra [13].

The first negative semantical result was obtained by Kerth [44]: he showed that no graph model satisfies the identity $\Omega_3 I = I$. This result shows a limitation of graph models, since the easiness of $\Omega_3 I$ was proven syntactically in [40] (see also [9]) and semantically in [1]; in this last article it is shown that for each closed t, there exists a filter model satisfying $\Omega_3 I = t$.

In this thesis we give a contribution to the the characterization of undefinedness in λ-calculus and to graph-easiness.

We define two sequences $\mathcal{M}_n$ and $\mathcal{G}_n$, $n \in \omega$, of infinite and recursive sets of mute terms, the *restricted regular mute* and *regular mute* terms respectively. Any term in $\mathcal{M}_n$ (resp. $\mathcal{G}_n$) has the form $s_0 s_1 \dots s_n$, for some n, where each s_i is an inductively built term called *restricted hereditarily n-ary* term (*hereditarily n-ary* term). Any n-ary restricted regular mute term $s \equiv s_0 s_1 \dots s_n$ has the property that, after n steps of head reduction, it reduces to a term of the shape $s_i t_1 \dots t_n$, where $1 \leq i \leq n$ and t_j are restricted hereditarily n-ary term. For regular mute terms a similar statement holds: any n-ary regular mute term $t \equiv s_0 s_1 \dots s_n$ has the property that, after n steps of head reduction, it reduces to a term of the shape $t_0 t_1 \dots t_n$, where all t_j are hereditarily n-ary term. This easily implies that restricted and regular mute terms are mute.

Furthermore, we show that sets $\mathcal{M}_n$ are graph-easy for any n. The starting point of the proof is a semantical property of graph models that only restricted regular mute have, thanks to their particular syntactical form. Then we apply a generalization of the forcing technique used in [13] to get, for any closed t, a graph model equating all elements of $\mathcal{M}_n$ to t.

2
Preliminaries

In this chapter we mainly follow [6].

2.1 The Lambda Calculus

The *Lambda-Calculus*, also denoted by λ-calculus, is a formal system consisting of a set of words called λ-*terms* over an alphabet and of a system of *rewriting* and *equating* rules.

Definition 2.1.1. *The set of λ-terms, denoted by Λ, is inductively built up from a denumerable set of variables* Var *according to the following rules:*

$$\begin{array}{llcl} (i) & x \in \mathrm{Var} & \Rightarrow & x \in \Lambda \\ (ii) & t, s \in \Lambda & \Rightarrow & (ts) \in \Lambda \\ (iii) & t \in \Lambda,\ x \in \mathrm{Var} & \Rightarrow & \lambda x.t \in \Lambda \end{array}$$

Any term (ts) obtained in rule (ii) is called *application* of the term t to s. Any term of the form $\lambda x.t$ is called *abstraction*: in it is is said that the variable x is under the scope of λx.

Any occurrence of the variable x is *free* if it is not in the scope of λx, otherwise it is *bound*. The set of free variables of a term t is denoted by $FV(t)$. A term t is *closed* if $FV(t) = \varnothing$. The set of closed λ-terms is denoted by Λ^0. Letters $t, s, p \dots$ usually range over elements of Λ while letters $x, y, z \dots$ usually range over the variables in Var. The symbol "$\equiv$" represents syntactical equality between λ-terms.

We follows Barendregt *variable convention*: in a set of terms T, then all bound variables are chosen to be different from the free variables. Therefore terms which differ only on the names of bound variables are called α-*equivalent* and are identified in the λ-calculus.

Important terms that will be used in the following are the identity $I \equiv \lambda x.x$, the projector on the first coordinate $K \equiv \lambda xy.x$, $S \equiv \lambda xyz.xz(yz)$, the Curry fixed point combinator $Y \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$, $\omega \equiv \lambda x.xx$ and $\Omega \equiv \omega\omega$, $\omega_3 \equiv \lambda x.xxx$ and $\Omega_3 \equiv \omega_3\omega_3$.

The basic equivalence relation on λ-terms is *convertibility*. In order to introduce it we need a substitution operator.

Definition 2.1.2. Convertibility *is defined by the following axioms and rules:*

$$
\begin{array}{lll}
(i) & t = t & \\
(ii) & t = s \;\Rightarrow\; s = t & \\
(iii) & t = s, s = p \;\Rightarrow\; t = p & \\
(iv) & t = s \;\Rightarrow\; tp = sp & \\
(v) & t = s \;\Rightarrow\; pt = ps & \\
(vi) & t = s \;\Rightarrow\; \lambda x.t = \lambda x.s & (rule\ \xi) \\
(vii) & (\lambda x.t)s = t[s/x] & (\beta\text{-}conversion)
\end{array}
$$

β-conversion is the axiom that characterizes the λ-calculus. It states that a term of the form $(\lambda x.t)s$, which is called *redex*, is equal to the term $t[s/x]$. The notation $t[s/x]$ represents the term obtained by replacing all free occurrences of x in t with s. Thanks to the variable convention this operation is well-defined.

The (β)-*reduction* is the result of the applications of the rule β from left to right only. It is denoted by $\to_\beta$. The reflexive, contextual and transitive closure of β-reduction is denoted by $\twoheadrightarrow_\beta$. Two terms t, s that are equivalent according to $\twoheadrightarrow_\beta$ (notation: $t =_\beta s$ or simply $t = s$) are called β-*equivalent* or β-*convertible*.

Definition 2.1.3. *The* Lambda calculus *is the formal system satisfying axioms* (i)-(vii).

Some syntactically different λ-terms may define the same function. Therefore the following axiom can be introduced.

$$(viii) \quad \lambda x.tx = t \quad \text{if } x \notin FV(t) \quad (\eta\text{-conversion})$$

If a formal system satisfies also $(viii)$ it is called *extensional λ-calculus* and denoted by $\lambda\beta\eta$-calculus. The reflexive, contextual and transitive closure of η-conversion is denoted by $\twoheadrightarrow_\eta$. Two terms equal according to $\twoheadrightarrow_\beta$ and $\twoheadrightarrow_\eta$ are called $\beta\eta$-*convertible*.

A *context* is a term where some variables are considered as holes denoted by [ ]. More formally:

(i) any variable x is a context.

(ii) [ ] is a context.

(iii) If $C_1[\ ]$ and $C_2[\ ]$ are contexts, then $C_1[\ ]C_2[\ ]$ and $\lambda x.C_1[\ ]$ are contexts.

(iv) If $C[\ ]$ is a context and t a term, then $C[t]$ denotes the term obtained by simultaneously replacing the "holes" with t. Variables are not renamed in this process.

It is possible to give a representation of natural numbers using λ-terms. There are many of such representations: the following one is the most commonly used.

Definition 2.1.4. Church's numerals. *Let f, x be variables. Given $n \in \omega$, the* Church numeral *of n is denoted by $\ulcorner n \urcorner$ and it is defined as follows:*

$$\lambda f \lambda x. \underbrace{f(f(\dots(f}_{n\ times}\, x)$$

So we have, for example, $\ulcorner 0 \urcorner \equiv \lambda f \lambda x.x$ and $\ulcorner 2 \urcorner \equiv \lambda f \lambda x.f(fx)$.

2.1.1 λI-calculus

Church originally defined a slightly different version of the Lambda Calculus, the λI-calculus,

Definition 2.1.5. *Λ_I is the set of terms of the λI-calculus. It is built according to the following rules:*

$$\begin{array}{lccc}
(i) & x \in \mathrm{Var} & \Rightarrow & x \in \Lambda_I \\
(ii) & t, s \in \Lambda & \Rightarrow & (ts) \in \Lambda_I \\
(iii) & t \in \Lambda \text{ and } x \in FV(t) & \Rightarrow & \lambda x.t \in \Lambda_I
\end{array}$$

2.1.2 Tree representation of λ-terms

Let Σ be an alphabet. Informally, a *labelled tree* is a tree that has at each node an element of Σ. We introduce a notion of tree based on the alphabet $\Sigma' = \{\lambda x_i, x_j\}$ where $i, j \in \omega$ and x_i, x_j are arbitrary variables.

Let t be a term. The *tree representation* $T(t)$ of t is defined inductively as follows:

- if x is any variable, then $T(x) = x$
- if t is an abstraction $\lambda x.s$, then

$$T(\lambda x.s) = \begin{tikzcd} \lambda x \arrow[d, dash] \\ T(s) \end{tikzcd}$$

- if t is an application sq, then

$$T(sq) = \begin{tikzcd} & \cdot \arrow[dl, dash] \arrow[dr, dash] & \\ T(s) & & T(q) \end{tikzcd}$$

2.2 Solvable and unsolvable terms

Definition 2.2.1. *A closed term t is called solvable if there exists an integer n and terms $s_1 \ldots, s_n$ such that $ts_1 \ldots s_n = I$. An arbitrary term t is solvable if one of its closures $\lambda \bar{x}.t$ is solvable.*

A term is called unsolvable if it is not solvable.

A λ-term has exactly one of the following forms:

(i) $\lambda x_1 \ldots x_n.yt_1 \ldots t_k$ $(n, k \geq 0)$;

(ii) $\lambda x_1 \ldots x_n.(\lambda y.s)ut_1 \ldots t_k$ $(n, k \geq 0)$.

It is said that the first term is in *head normal form* (hnf, for short). The redex $(\lambda y.s)u$ in the second one is called *head redex.* If a term t is β-equivalent to a term s in head normal form we say that it has a head normal form.

A step of β-reduction that reduces the head redex is denoted by $\rightarrow_h$. A reduction strategy that reduces at each step the head redex of a term and stops if there is no such redex is called *head reduction.* The term obtained from a term t after a terminating head reduction is called *principal head normal form of t.* It can be proved that a term t has a hnf iff the head reduction of t terminates.

An unsolvable term t has:

(i) order 0 if it is not β-equivalent to an abstraction;

(ii) order ∞ if, for every natural number $n > 0$, $t =_{\lambda\beta} \lambda x_1 \dots x_n.u$ for some u;

(iii) order $n \geq 1$ if there exists a greatest positive number n such that $t =_{\lambda\beta} \lambda x_1 \dots x_n.u$ for some u.

For example, Ω has order 0, $\lambda x.\Omega$ has order 1 and $\mathbf{Y}K$ has order ∞. Terms of order 0 are also called *zero terms.*

Theorem 2.2.2. *(Wadsworth) [6], p.41) A term t is solvable iff it has a head normal form.*

So, from now on we call solvable (resp. unsolvable) terms with (without) head normal form. Another result by Wadsworth states that if t has a head normal form, then the head reduction path of t stops after a finite number of steps ([6], p.177). This implies that the set of unsolvables is co-recursively enumerable.

2.2.1 Böhm trees

Let $\Sigma = \{\perp\} \cup \{\lambda x_1, \dots, x_n.y :\ \text{where } x_i, y \text{ are variables and } i \in \omega\}$. The labelled tree of t, called the *Böhm tree* of t and denoted by $BT(t)$, is informally defined as follows.

- if the principal head normal form of t is $\lambda x_1 \dots \lambda x_n.ys_1 \dots s_k$, then

$$BT(t)= \begin{tikzcd} & \lambda x_1, \dots \lambda x_n.y \arrow[dl, dash] \arrow[dr, dash] & \\ BT(s_1) & \cdots & BT(s_k) \end{tikzcd}$$

- otherwise, $BT(t) = \perp$

This is not a formal definition: given a term t whose principal head normal form is $\lambda x_1 \dots \lambda x_n.yt_1 \dots t_q$, one of the t_i may be more complex than t itself ([6] p. 216).

2.3 λ-theories

Definition 2.3.1. *A set of equations between λ-terms is a λ-theory if it is an equivalence relation and it is closed under the rules of Definition 2.1.2.*

$T \vdash t = s$ denotes that the equation $t = s$ belongs to T.

A theory T is *inconsistent* if it equates any pair of terms. Otherwise, it is *consistent*Given an arbitrary set of equations Σ, there always exists the smallest theory T containing Σ: we say that T *is generated by* Σ. The smallest λ-theory is denoted by $\lambda\beta$-theory.

A theory is *extensional* if it is closed also by axiom $(viii)$. The smallest extensional λ-theory is denoted by $\lambda\beta\eta$-theory.

There exists a continuum of consistent λ-theories . The theory generated by equating all unsolvables is denoted by $\mathcal{H}$. Theories that contains $\mathcal{H}$ are called *sensible*. $\mathcal{H}$ admits a unique maximal extension $\mathcal{H}^*$. A λ-theory is *semi-sensible* if it never equates a solvable with an unsolvable. It can be shown that every sensible theory is also semi-sensible. (see ([6]), chp. 16, for a survey on λ-theories).

2.4 Denotational semantics

Given sets X, Y, we write $X \subseteq_f Y$ if X is a finite subset of Y.

2.4.1 Categorical models

Definition 2.4.1. *A category $\mathbb{C}$ is* cartesian closed *(ccc for short) if the following conditions hold:*

- *there is an object $\top \in \mathbb{C}$ such that, for every $A \in \mathbb{C}$, there is exactly one $f \in \mathbb{C}(A, \top)$. $\top$ is called* terminal object.

- *For every couple of objects A_1, A_2 there exists an object $A_1 \times A_2$, called* cartesian product, *and arrows $\pi_i \in \mathbb{C}(A_1 \times A_2, A_i)$, $(i = 1, 2)$, called* projections, *satisfying the following property: for every couple of arrows $f_i \in \mathbb{C}(C, A_i)$, $(i = 1, 2)$ there exists a unique $\langle f_1, f_2\rangle \in \mathbb{C}(C \to A_1 \times A_2)$ such that $f_i = \pi_i \circ \langle f_1, f_2\rangle$.*

- *Given $A, B \in \mathbb{C}$, there is an an object $A \Rightarrow B$ called* exponent *and an arrow* $\mathrm{Ev} \in \mathbb{C}((A \Rightarrow B) \times A, B)$ *such that for every $f \in \mathbb{C}(C \times A, B)$ there is a unique $\Lambda(f) \in \mathbb{C}(C, A \Rightarrow B)$ satisfying the equation*

$$f = \mathrm{Ev} \circ (\Lambda(f) \times Id_A)$$

where Id_A is the identity arrow on A.

A *reflexive object* in a ccc $\mathbb{C}$ is a triple $\mathcal{U} = (U, \mathrm{App}, \mathrm{Lam})$, where $U \in \mathbb{C}$, and arrows $\mathrm{Lam} \in \mathbb{C}(U \Rightarrow U, U)$ and $\mathrm{App} \in \mathbb{C}(U, U \Rightarrow U)$ satisfying

$$\mathrm{App} \circ \mathrm{Lam} = Id_{U \Rightarrow U}.$$

If also $\mathrm{Lam} \circ \mathrm{App} = Id_U$ holds, $\mathcal{U}$ is called *extensional*.

For any object $U \in \mathbb{C}$, we define $\top = U^0$ and $U^{n+1} = U^n \times U$. Given a finite subset $I = \{x_1, \ldots, x_n\}$ of Var, we set $U^I = U^n$ and $\pi^I_{x_i} \in \mathbb{C}(U^I, U)$, the projection arrow on the coordinate i. If $I \subseteq_f$ Var contains all free variables of a λ-term t, we say that I is an *adequate set* for t.

Let $\mathcal{U} = (U, \mathrm{App}, \mathrm{Lam})$ be a reflexive object, t a λ-term and I an adequate set for t. We define by induction on the complexity of t an object $|t|^{\mathcal{U}}_I$ in $\mathbb{C}(U^I, U)$ called the *interpretation* of t:

- $|x|^{\mathcal{U}}_I = \pi_{x_i}$, for any variable x_i.
- $|st|^{\mathcal{U}}_I = \mathrm{Ev} \circ \langle \mathrm{App} \circ |s|^{\mathcal{U}}_I, |t|^{\mathcal{U}}_I \rangle$, for any application term st.
- $|\lambda y.t|^{\mathcal{U}}_I = \mathrm{Lam} \circ \Lambda(|t|^{\mathcal{U}}_{I \cup \{y\}})$, for any abstraction term $\lambda y.t$ (w.l.o.g we suppose $y \notin I$).

If it is clear from the context, the superscript $^{\mathcal{U}}$ is dropped in $|t|^{\mathcal{U}}_I$: if t is closed also the subscript $_I$ is dropped.

In a categorical model $\mathcal{U}$ terms with the same interpretation are identified. More formally, given a reflexive object $\mathcal{U}$, we can define the *equational theory* $Th_=(\mathcal{U})$ of $\mathcal{U}$:

$$Th_=(\mathcal{U}) = \{t = s \ : \ |t|^{\mathcal{U}}_I = |s|^{\mathcal{U}}_I, \text{ where } I = FV(t) \cup FV(s)\}$$

In $Th_=(\mathcal{U})$, β-equivalent terms are always identified, i.e. $Th_=(\mathcal{U})$ is always a λ-theory.

2.4.2 Scott-continuous models

The first model of the λ-calculus was found in late 1960s by Dana Scott. He built an algebraic lattice D_∞ isomorphic to $D_\infty^{D_\infty}$, the set of all continuous functions from D_∞ to itself, according to a particular topology called *Scott topology*.

This model is only one example of a general kind of categorical models that live in the category of complete partial orders with Scott continuous functions.

Let $\mathcal{D} = (D, \sqsubseteq)$ be a partially order set (*poset* for short). Two elements x, y of D are *compatible* if there is $z \in D$ such that $x \leq z$ and $y \leq z$ (z is an upper bound of x and y). A non-empty subset X of D is *directed* if every pair of elements $x, y \in X$ has an upper bound in X.

Definition 2.4.2. *A poset $\mathcal{D}$ is a* complete partial order *(or cpo for short) if:*

- *it has a least element $\bot$, i.e. for every $x \in D$ $\bot \leq x$.*

- *Every directed subset X of D has a least upper bound, denoted by $\sqcup X$, belonging to D.*

A poset such that every $X \subseteq D$ has a least upper bound is a complete lattice*.*

An element d of a poset $\mathcal{D}$ is *compact* if for any directed $X \subseteq D$, if $d \leq \sqcup X$, then there is $x \in X$ such that $d \leq x$, A cpo is *algebraic* if for every element d of D, the set $C = \{x \in D : x \leq d \text{ and } x \text{ is compact}\}$ is directed and $d = \sqcup C$

Definition 2.4.3. *The* Scott topology *on a poset $\mathcal{D}$ is the collection of all sets $O \subseteq D$ such that:*

- *$x \in O$ and $x \leq y$ implies $y \in O$.*
- *X is directed and $\sqcup X \in O$ implies $X \cap O \neq \varnothing$.*

In Scott topology continuous functions have a strong characterization in terms of directed sets.

Let D and D' be cpos. Then a function $f : D \to D'$ is continuous iff

$$f(\sqcup X) = \sqcup f(X), \text{ for all directed sets } X \subseteq D.$$

The category whose objects are cpos and arrows are continuous functions between cpos is denoted by **CPO**. It can be proved that **CPO** is a ccc. A *Scott-continuous* λ-model is a reflexive object of **CPO**.

2.4.3 Graph models

Among Scott-continuous models of the λ-calculus, graph models are arguably the simplest.

The first graph model $\mathcal{P}_\infty$ was introduced in the early 1970s independently by Plotkin and Scott . Soon afterwards Plotkin and Engler introduced $\mathcal{E}$, an even simple graph model.

Graph models are very simple structures because they can be described by using just a set and a function. In the following we denote by D^* the set of all finite subsets of D.

Definition 2.4.4. *A graph model is a pair $\mathcal{D} = (D, p)$, where D is a denumerable set, called the web of $\mathcal{D}$, and $p : D^* \times D \to D$ is an injective function.*

Such a pair is called *total pair*. In the setting of graph models a *partial pair* is a pair (A, q) where A is any set and $q : A^* \times A \rightharpoonup A$ is a partial (possibly total) injection.

If (D, p) is a partial pair, we sometimes write $a \to_p \alpha$ (or $a \to \alpha$ if p is evident from the context) for $p(a, \alpha)$. Moreover, $\beta \to \alpha$ means $\{\beta\} \to \alpha$. The notation $a_1 \to a_2 \to \cdots \to a_{n-1} \to a_n \to \alpha$ stands for $(a_1 \to (a_2 \to \ldots (a_{n-1} \to (a_n \to \alpha)) \ldots))$. If $\bar{a} = a_1, a_2, \ldots, a_n$, then $\bar{a} \to \alpha$ stands for $a_1 \to a_2 \to \ldots a_{n-1} \to a_n \to \alpha$.

Given a graph model $\mathcal{D} = (D, p)$, an *environment* is a function $\rho : \mathrm{Var} \to \mathcal{P}(D)$. The set of all environments of $\mathcal{D}$ is denoted by $\mathrm{Env}_{\mathcal{D}}$.

Definition 2.4.5. *A total pair (D,p) generates a λ-model whose universe is $\mathcal{P}(D)$, called graph λ-model. The interpretation $|t|^p : \mathrm{Env}_{\mathcal{D}} \to \mathcal{P}(D)$ of a λ-term t in the graph model (D,p) can be described inductively as follows:*

- $|x|^p_\rho = \rho(x)$.
- $|tu|^p_\rho = \{\alpha : (\exists a \subseteq_f |u|^p_\rho)\ \ a \to \alpha \in |t|^p_\rho\}$.
- $|\lambda x.t|^p_\rho = \{\, a \to \alpha : \alpha \in |t|^p_{\rho[x:=a]}\}$.

Since $|t|^p_\rho$ only depends on the value of ρ on the free variables of t, we write $|t|^p$ if t is closed.

A graph model (D,p) *satisfies* $t = u$, written $(D,p) \vDash t = u$, if $|t|^p_\rho = |u|^p_\rho$ for all environments ρ. The λ-theory $\mathrm{Th}(D,p)$ induced by (D,p) is defined as

$$\mathrm{Th}(D,p) = \{t = u : t, u \in \Lambda \text{ and } |t|^p_\rho = |u|^p_\rho \text{ for every } \rho\}.$$

A λ-theory induced by a graph model is called a *graph theory*.

2.4.4 Forcing in graph models

The forcing technique in graph models was introduced by Baeten and Boerboom in [4], where they used this technique to build, for any term t, a graph model (D,p) such that $(D,p) \models t = \Omega$. By imposing some conditions on the function p in the graph model (D,p), they force some elements of D to belong to the interpretation of Ω: that is where the name *forcing* comes from.

The following proposition clarifies what we have just said. It gives a necessary condition and a sufficient one for an element to be in the interpretation of Ω in a graph model.

Proposition 2.4.6. *([4]) Let (D,p) be a graph model and $\alpha \in D$.*

1. *If $\alpha \in |\Omega|^p$, then there exists a finite subset a of D such that $p(a,\alpha) \in a$;*
2. *If there exists β such that $\beta = p(\{\beta\},\alpha)$, then $\alpha \in |\Omega|^p$.*

Once the conditions on p, matching a given purpose, have been found, the following step is to build a graph model satisfying those conditions; this is achieved by starting from a suitable partial pair, and then carefully completing it to a total pair. In [4], this is achieved with an *ad hoc* construction.

A generalization of this construction, involving a notion of weakly continuous function and presented below as Theorem 2.4.8, has been proposed in [13]. In the following we report the main points of this technique.

Notation 1. *Let D be an infinite countable set. By $\mathcal{I}(D)$ we indicate the cpo of partial injections $q : D^* \times D \rightharpoonup D$, ordered by inclusion of their graphs.*

By a "*total q*" we will mean "an element of $\mathcal{I}(D)$ which is a total map" (equivalently: which is a maximal element of $\mathcal{I}(D)$). The domain and range of $q \in \mathcal{I}(D)$ are denoted by $dom(q)$ and $rg(q)$. We will also confuse the partial injections and their graphs.

Definition 2.4.7. *[13, Definition 10] A function $F : \mathcal{I}(D) \to \mathcal{P}(D)$ is weakly continuous if it is monotone with respect to inclusion and if furthermore, for all total $p \in \mathcal{I}(D)$,*

$$F(p) = \bigcup_{q \subseteq_{\text{fin}} p} F(q).$$

Since we are working with a countable infinite D, the difference with continuity comes of course from the fact that there exist elements of $\mathcal{I}(D)$ which are not total but of infinite cardinality.

The forcing completion process we were referring to is the core of the proof of Theorem 2.4.8 below, which is the fundamental tool to prove the graph easiness of Ω in [13].

The next lemma is an application of the notion of forcing.

Theorem 2.4.8. *[13, Theorem 11] If $F : \mathcal{I}(D) \to \mathcal{P}(D)$ is weakly continuous, then there exists a total p such that $|\Omega|^p = F(p)$.*

In order to introduce the notion of forcing we need the following definition,

Definition 2.4.9. *The set Λ_D of* generalized λ-terms *w.r.t. a denumerable set D is the smallest set satisfying the following conditions:*

- $\mathrm{Var} \subseteq \Lambda_D$.
- $\mathcal{P}(D) \subseteq \Lambda_D$.
- $t, s \in \Lambda_D$ *implies* $ts \in \Lambda_D$.
- $t \in \Lambda_D$ *and* $x \in \mathrm{Var}$, *then* $\lambda x.t \in \Lambda_D$.
- *if $f : \mathcal{P}(D)^n \to \mathcal{P}(D)$ is a continuous function of arity $n \geq 1$ w.r.t. the cpo $(\mathcal{P}(D), \subseteq)$ and $t_1, \ldots, t_n \in \Lambda_D$, then $f(t_1, \ldots, t_n) \in \Lambda_D$.*

Definition 2.4.10. *[13, Definition 14]*(Forcing). *Given t a closed term of Λ_D, $q \in \mathcal{I}(D)$ and $\alpha \in D$, the abbreviation $q \Vdash \alpha \in t$ means that, for all total injections $p \supseteq q$, we have that $(D, p) \models \alpha \in |t|^p$. Furthermore $q \Vdash_\rho Y \subseteq t$ means that $q \Vdash \alpha \in |t|$ for all $\alpha \in Y$.*

The next lemma is an application of the notion of forcing.

Lemma 2.4.11. *[13, Lemma 15] For every closed λ-term t, the function $F_t : \mathcal{I}(D) \to \mathcal{P}(D)$, defined by $F_t(q) = \{\alpha \in D \ : \ \forall$ total $p \supseteq q,\ \alpha \in |t|^p\}$, is weakly continuous, and we have $F_t(p) = |t|^p$ for each total p.*

Graph easiness of Ω is a simple corollary of Lemma 2.4.11 and Theorem 2.4.8.

3

Undefinedness in lambda calculus

The problem of characterizing λ-terms that represent an undefined computational process has interested researchers since the origin of λ-calculus. This problem is difficult to formalize and consequently it is not clear how to give a definite answer to it. In this chapter we give a short survey of this topic. [6], pp. 39-43., is the main reference of this chapter, especial for Sections 3.1 and 3.2.

3.1 λ-definability of partial functions

The issue of terms representing undefined processes naturally arises when considering the problem of representation of partial functions. I recall here the classic characterization of a computable function g:

$$g(n) = \begin{cases} \text{the result of the effective procedure computing } g & \text{if } n \text{ belongs to the domain of } g; \\ \text{undefined} & \text{otherwise.} \end{cases}$$

So the natural definition of λ-representable function seems to be the following one.

Definition 3.1.1. *Given a partial function $f : \omega^p \to \omega$, we say that f is λ-representable if there exists a λ-term F such that, for any p-uple of natural numbers $n_1, \ldots, n_p$,*

$$F\lceil n_1 \rceil \ldots \lceil n_p \rceil = \begin{cases} \lceil f(n_1, \ldots, n_p) \rceil & \textit{if } (n_1, \ldots, n_p) \textit{ belongs to } Dom(f); \\ \textit{a non-normalizing term} & \textit{otherwise.} \end{cases}$$

When dealing with a total function f this definition is completely satisfactory: the term F suitably represents the function f in the λ-calculus.

Instead, if we want to represent a proper partial function g with a term G, it is not clear how to characterize the term $G\lceil n \rceil$ when n does not belong to the domain of g. It is reasonable that $G\lceil n \rceil$ must not have a normal form, but it is not obvious that this condition is also sufficient.

3.2 Undefined terms in λK-calculus

The solution given by Definition 3.1.1 was proposed by Church for the λI calculus.

Moreover, he suggested the following general identification: a term represents an undefined process iff it does not have a normal form.

In λI calculus this statement has some desirable properties.

- The theory $\mathbf{K} = \{(t, s) : t \text{ and } s \text{ do not have a normal form}\}$ is consistent in λI calculus ([6], p. 416).
- A term t is solvable iff it has a normal form ([6], p. 42).

In the λK-calculus the above identification cannot be accepted. In fact the properties listed above do not hold anymore .

- The theory $\mathbf{K} = \{(t, s) : t \text{ and } s \text{ do not have a normal form}\}$ is inconsistent in λ calculus ([6] p.39).
- There exist solvable terms that do not have a normal form.

The first property expresses the fact that the notion "without normal form" is too syntactical.

These facts prevent us from accepting the identification proposed by Church in the λK-calculus.

3.2.1 "Undefined = unsolvable"

Barendregt and Wadsworth proposed ([6], p. 42) the following identification: A term represents an undefined process iff it is unsolvable.

The definition of representable function is changed accordingly:

Definition 3.2.1. *Given a partial function $f : \omega^p \to \omega$, we say that f is* strongly representable *if there exists a λ-term F such that, for any p-uple of natural numbers $n_1, \ldots, n_p$,*

$$F\lceil n_1 \rceil \ldots \lceil n_p \rceil = \begin{cases} \lceil f(n_1, \ldots, n_p) \rceil & \textit{if } (n_1, \ldots, n_p) \textit{ belongs to } Dom(f); \\ \textit{an unsolvable term} & \textit{otherwise.} \end{cases}$$

Unsolvable terms have properties that "undefined" terms should have (in particular they satisfy the properties that terms without normal form have in λI calculus):

- the theory $\mathcal{H} = \{(t, s) : t, s \text{ are unsolvable}\}$ is consistent.
- Unsolvable terms are interpreted as the bottom element $\perp$ in Scott's models.
- A term is unsolvable iff it does not have head normal form.

Perhaps, the strongest argument in favour of this proposal is the following genericity lemma, which shows the lack of computational meaning of the unsolvable.

Proposition 3.2.2. *(Genericity Lemma) Let t be an unsolvable and s a normal form. Then, for any context $C[\ ]$,*

$$C[t] = s \ \Rightarrow\ \forall p \in \Lambda,\ C[p] = s.$$

Thanks to all these considerations, the proposed identification is commonly accepted. If the operational semantics of a λ-term is its Böhm tree it is natural that the terms representing undefinedness are the unsolvable terms.

We will see in the following that it is possible to introduce new criteria of undefinedness which allow to isolate other sets of terms.

3.3 A fine classification of undefinedness

We may classify the order of undefinedness of a set of closed λ-terms according to the "size" of the set of terms it can be consistently equated to.

3.3.1 Easiness

Definition 3.3.1. *Let Y be a set of closed λ-terms. We say that the set Y is* consistent *if the theory $\Sigma = \{t = u : t \in Y\}$ is consistent. We define*

$$\mathrm{Con}(Y) = \{u \in \Lambda^0 : Y \cup \{u\} \textit{ is consistent}\}.$$

If $Y = \{t\}$ is a singleton set, we write $\mathrm{Con}(t)$ for $\mathrm{Con}(\{t\})$. It is obvious that $Y \subseteq \Lambda^0$ is consistent if and only if $\mathrm{Con}(Y) \neq \emptyset$.

Definition 3.3.2. *A set $Y \subseteq \Lambda^0$ is an* easy-set *if* $\mathrm{Con}(Y) = \Lambda^0$.

A term t is called *easy* if $\{t\}$ is an easy-set.

Easy terms were introduced in [39], where it is also shown using syntactical techniques that Ω is easy. Semantical proofs of easiness originated in [4].

In the following definition we introduce the notion of $\mathcal{C}$-easiness for a class $\mathcal{C}$ of models of λ-calculus.

Definition 3.3.3. *Let $Y \subseteq \Lambda^0$ and $\mathcal{C}$ be a class of models of λ-calculus. We say that Y is a $\mathcal{C}$-*easy set *if, for every closed λ-term t, there exists a model $\mathcal{N} \in \mathcal{C}$ in which all elements of $Y \cup \{t\}$ have the same interpretation. A λ-term t is called $\mathcal{C}$-easy if $\{t\}$ is a $\mathcal{C}$-easy set.*

A term t is *nf-easy* if $\mathrm{Con}(t) \supseteq \{u \in \Lambda^0 : u \text{ is a normal form}\}$.

The set of easy terms is a proper co-recursively enumerable subset of the unsolvables. For example, Ω_3 is unsolvable but not easy, because it cannot be consistently equated to the identity I. Although Ω_3 is not easy, it is possible to show that $\Omega_3 I$ is easy ([40]).

In [9] Berarducci and Intrigila prove many interesting results on easy terms. Some of them are collected in the following theorem that shows the unusual behavior of easy terms.

Theorem 3.3.4. *[9] The following conditions hold:*

1. *There exists* $t \in \Lambda^0$ *such that* $\mathrm{Con}(t) = \Lambda^0 \setminus [I]_{\lambda\beta}$*, where* $[I]_{\lambda\beta}$ *is the set of terms that are* β*-equivalent to the identity* I*.*

2. *There exists a nf-easy term that is not easy.*

3. *A term* t *is easy iff* $\mathrm{Con}(t) \supseteq \{u \in \Lambda^0 : \mathrm{BT}(u) \text{ is finite}\}$*.*

4. $u \in \mathrm{Con}(\mathbf{Y}\Omega_3)$ *for every term* u *such that* $BT(u)$ *is not a subtree of* $BT(\omega_3)$*.*

Any element of an easy-set is obviously an easy term. Berline-Salibra [13] have shown that the infinite set $(\{\Omega(\lambda x_0 \dots x_k.x_k) : k \in \omega\}$ is an easy-set. There exist sets of easy terms that are not easy-sets: easiness of $\{\Omega, \Omega I\}$ fails because $\{\Omega, \Omega I, K\}$ is not consistent. In particular, the set of all easy terms is not an easy-set ([38]).

In general, proofs of easiness are difficult. For example, it is not yet known whether $\mathbf{Y}\Omega_3$ is easy or not ([14]).

3.3.2 Statman-sets

In this section we present the most suitable candidates for representing the undefined value of a partial recursive function in the λ-calculus.

Definition 3.3.5. *A* β*-closed set* $B \subseteq \Lambda^0$ *is a* Statman-set *if, for every recursive partial function* $f : \omega \to \omega$*, there exists* $F \in \Lambda^0$ *such that*

$$\begin{cases} F\ulcorner n \urcorner =_\beta \ulcorner f(n) \urcorner & \text{if } f \downarrow n; \\ F\ulcorner n \urcorner \in B & \text{otherwise.} \end{cases}$$

Statman has shown the following result in an unpublished paper [62]. The proof by Statman is based on early results by Visser [68] and can be found in [7].

Theorem 3.3.6. *[7, Theorem 4.1] Every nonempty co-recursively enumerable* β*-closed set of closed* λ*-terms is a Statman-set.*

As a trivial consequence, and using that $\mathrm{Con}(A)$ is a co-recursively enumerable set for every β-closed recursively enumerable set A, we get the following proposition.

Proposition 3.3.7. *Let* $A \subseteq \Lambda^0$ *be a* β*-closed recursively enumerable set such that* $A \neq \Lambda^0$*. Then we have:*

1. *$\Lambda^0 \setminus A$ is a Statman-set.*

2. *$\mathrm{Con}(A)$ is a Statman-set for every $\mathrm{Con}(A) \neq \emptyset$. In particular, $\mathrm{Con}(t)$ is a Statman-set for every closed λ-term t.*

Example 3.3.8. The set of closed λ-terms without normal form is a Statman-set. The same holds for the set of unsolvable (resp. easy, zero) closed terms.

3.3.3 Mute terms

Mute terms were introduced in [8] by Berarducci. All results and definitions of this section can be found in [8].

Berarducci trees take in account the computational content of the unsolvables. As for Böhm trees, Berarducci trees are obtained by an infinite unfolding of λ-terms.

A *top normal form* (top-nf, for short) is either a variable or an abstraction or a zero term applied to another term. A term t has a top-nf if t is β-convertible to a top-nf.

The Berarducci tree of a term t is the possibly infinite unfolding of t according to the following coinductive definition:

$$\mathrm{BD}(t) = \begin{cases} \perp & \text{if } t \text{ has no top-nf;} \\ x & \text{if } t =_\beta x. \\ \lambda x.\mathrm{BD}(u) & \text{if } t =_\beta \lambda x.u. \\ \mathrm{BD}(s) \cdot \mathrm{BD}(u) & \text{if } t =_\beta su \text{ with } s \text{ zero-term.} \end{cases}$$

The function BD is well-defined by [8, Theorem 9.5].

As an example, we build the Berarducci tree of the unsolvable term Ω_3. The only possible reduction path of Ω_3 is the following one:

$$\Omega_3 \to_\beta \Omega_3\omega_3 \to_\beta \Omega_3\omega_3\omega_3 \to_\beta \Omega_3\omega_3\omega_3\omega_3 \to_\beta \dots$$

where $\omega_3 \equiv \lambda x.xxx$. So the zero term Ω_3 can be seen as an infinite term, namely $(((\dots)\omega_3)\omega_3)\omega_3$.

One of the main results of [8] is that the λ-theory $\mathrm{BD} = \{t = u : \mathrm{BD}(t) = \mathrm{BD}(u)\}$ of Berarducci trees is consistent. The terms that have a bottom Berarducci tree are called mute terms and can be formally defined as follows.

Definition 3.3.9. *[8] A term t is mute if t has no top-nf.*

Mute terms have a totally undefined operational behavior. By developing a confluent extension of the λ-calculus, Berarducci proved that they satisfy the strongest property of undefinedness we have introduced so far, namely.

Theorem 3.3.10. *[8] The set of mute terms is an easy-set.*

We have that being a zero term is a co-recursively enumerable property. By Definition 3.3.9, the set of mute terms is not recursively enumerable nor co-recursively enumerable.

4

The Regular mute terms

4.1 Introduction

In this chapter we introduce *regular mute* terms. Regular mute terms are mute terms built with an inductive definition. This is in strong opposition to classical mute terms, that instead are defined as terms that satisfy negative conditions.

We had the intuition for introducing regular mutes in the "replication behavior" that the typical mute term Ω has. In fact, we prove that a regular mute, after a finite number of head reductions, has the form of a regular mute once again. This result easily proves that regular mutes are mutes, so they share the same strong properties of undefinedness mutes have: they are unsolvable, easy and the set of regular mute is an easy set.

In the second part of the chapter we prove that the a subclass of regular mutes, called *restricted* regular mute, is a countable union of graph-easy sets (3.3.3). Restricted regular mutes are defined by imposing a restrictive syntactical condition to the main definition. Thanks to their simpler form, they satisfy some various technical lemmas which allow us to prove graph-easiness. The main technical tool used here is an application of the forcing technique to graph models.

4.2 Regular mute terms

4.2.1 Hereditarily n-ary terms

In order to introduce regular mute terms, we introduce hereditarily n-ary terms.

Definition 4.2.1. *Let V be the infinite set of variables of λ-calculus and $n \geq 1$. The set $H_n[V]$ of* restricted hereditarily n-ary terms *(over V) is the smallest set of λ-terms containing V and such that: for all $t_1, \ldots, t_n \in H_n[V]$, distinct variables $y_1, \ldots, y_n \in V$ and $i \leq n$ we have: $\lambda y_1 \ldots y_n.y_i t_1 \ldots t_n \in H_n[V]$.*

The set $K_n[V]$ of hereditarily n-ary terms *(over V) is the smallest set of λ-terms containing V and such that: for all $t_1, \ldots, t_n \in K_n[V]$, distinct variables $y_1, \ldots, y_n \in V$ and $z \in V \cup \bar{y}$, we have: $\lambda y_1 \ldots y_n.z t_1 \ldots t_n \in K_n[V]$.*

We denote by $H_n[\bar{x}]$, for $\bar{x}$ any finite (and possibly empty) sequence of distinct variables in V, the set of terms of $H_n[V]$ whose free variables are included in $\bar{x}$. Similarly, we denote by $K_n[\bar{x}]$ the set of terms of $K_n[V]$ whose free variables are included in $\bar{x}$.

To simplify the notation, we write H_n for $H_n[\,]$ and K_n for $K_n[\,]$.
Notice that:

- $t \in H_n[\bar{x}]$ iff either t is a variable in $\bar{x}$ or there exists a sequence $\bar{y}$ of distinct variables such that $t \equiv \lambda y_1 \dots y_n.y_i t_1 \dots t_n$, where $t_j \in H_n[\bar{x}, \bar{y}]$.
- $t \in K_n[\bar{x}]$ iff either t is a variable in $\bar{x}$ or there exists a sequence $\bar{y}$ of distinct variables such that $t \equiv \lambda y_1 \dots y_n.z t_1 \dots t_n$, where $t_j \in K_n[\bar{x}, \bar{y}]$ and $z \in \bar{x} \cup \bar{y}$.

Example 4.2.2. Some unary and binary hereditarily λ-terms:

- $\lambda x.xx \in H_1$.
- $\lambda y.yx \in H_1[x]$.
- $\lambda x.x(\lambda y.yx) \in H_1 \quad (\lambda y.yx \in H_1[x])$.
- $\lambda zy.yzx \in H_2[x]$.
- $\lambda xy.x(\lambda zt.tzx)y \in H_2 \quad (\lambda zt.tzx, y \in H_2[x, y])$.
- $\lambda x.yx \in K_1[y]$.
- $\lambda y.y(\lambda x.yx) \in K_1 \quad (\lambda x.yx \in K_1[y])$.

Given a natural number n and variables $\bar{x}$ we define inductively a sequence of sets of λ-terms:

Definition 4.2.3. *Let $\bar{x} = x_1, \dots x_k$ and $\bar{y} = y_1, \dots, y_n$ be distinct variables. We define:*

- $K_n^0[\bar{x}] = K_n[\bar{x}]$
- $K_n^{m+1}[\bar{x}] = \{s[\overline{u}/\overline{y}] : s \in K_n^m[\bar{x}, \bar{y}], \bar{u} = u_1, \dots, u_n \in K_n^m[\bar{x}]\}$

By using sets $K_n^m[\bar{x}]$, we define

$$T_n[\bar{x}] = \bigcup_{m \in \omega} K_n^m[\bar{x}].$$

The *rank* of a term $t \in T_n[\bar{x}]$ is the smallest natural m such that $t \in K_n^m[\bar{x}]$; it is denoted by $rk(t)$.

Example 4.2.4. By Example 4.2.2 we have $\lambda x.yx \in K_1[y]$ and $\lambda z.zz \in K_1$. So $\lambda x.(\lambda z.zz)x \in K_1^1$.

4.2.2 Syntactical properties of hereditarily n-ary terms

In the following, if $\bar{x}$ is a sequence, then $l(\bar{x})$ denotes its length.

Lemma 4.2.5. (Closure of $H_n[V]$ under substitution) *Let $n \in \omega$ and $t \in H_n[V]$. Then:*

(i) *For all $\bar{z} \in V$ and all $\bar{u} \in H_n[V]$ such that $l(\bar{u}) = l(\bar{z})$, we have $t[\bar{u}/\bar{z}] \in H_n[V]$.*

(ii) *Moreover if $t \in H_n[\bar{x}, \bar{z}]$ and $\bar{u} \in H_n[\bar{x}]$, then $t[\bar{u}/\bar{z}] \in H_n[\bar{x}]$.*

Proof. (i) Induction on the complexity of t.

- If $t \equiv x$ is a variable, then $x[\bar{u}/\bar{z}]$ is equal to x (if $x \notin \bar{z}$) or to u_i (if $x = y_i$).
- If $t \equiv \lambda y_1 \dots y_n.y_i t_1 \dots t_n$, we can suppose w.l.o.g that every y_i is not a free variable in any $u_j \in \bar{u}$. Then $t[\bar{u}/\bar{z}] \equiv \lambda y_1 \dots y_n.y_i t_1[\bar{u}/\bar{z}] \dots t_n[\bar{u}/\bar{z}]$. By ind. hyp., $t_i[\bar{u}/\bar{z}] \in H_n[V]$, so also $t[\bar{u}/\bar{z}] \in H_n[V]$.

(ii) Trivial consequence of (i).

□

Lemma 4.2.6. *1. Given $n \geq 1$, if $\bar{x} \subseteq \bar{y}$ and $m \leq p$, then $H_n[\bar{x}] \subseteq H_n[\bar{y}]$ and $K_n^m[\bar{x}] \subseteq K_n^p[\bar{y}]$. Inclusions are strict iff $\bar{x} \subsetneq \bar{y}$ or $m < p$.*

2. A term $t \in K_n$ has form $\lambda \bar{y}.y_i t_1 \dots t_n$, but in general it is not a restricted hereditarily n-ary term.

Proof. 1. First we prove that $\bar{x} \subseteq \bar{y}$ implies $H_n[\bar{x}] \subseteq H_n[\bar{x}]$, by induction on the length of t.

- If $t \equiv z$ is a variable, by definition $z \in H_n[\bar{x}]$ iff $z \in \bar{x}$. This implies that $z \in H_n[\bar{y}]$.
- If t is not a variable, then there exists a sequence $\bar{z}$ of distinct fresh variables such that $t \equiv \lambda z_1 \dots z_n.z_i t_1 \dots t_n$, with $t_j \in H_n[\bar{x}, \bar{z}]$. By ind. hyp., all t_j are in $H_n[\bar{y}, \bar{z}]$, so also $t \in H_n[\bar{y}]$.

Similarly we prove that $K_n[\bar{x}] \subseteq K_n[\bar{y}]$. By Definition 4.2.3, this easily implies that $K_n^m[\bar{x}] \subseteq K_n^m[\bar{y}]$. By construction $K_n^m[\bar{x}] \subseteq K_n^p[\bar{x}]$ when $m \leq p$.

If $z \in \bar{y} \setminus \bar{x}$, then $z \in K_n[\bar{y}] \setminus K_n[\bar{x}]$, so $z \in K_n^m[\bar{y}] \setminus K_n^m[\bar{x}]$ for any $m \geq 1$. For any $n \geq 1$, we have that $s \equiv \lambda x_1 \dots x_n.y x_1 \dots x_n \in K_n[y]$ and $t_0 \equiv \lambda x_1 \dots x_n.x_1 x_1 \dots x_n \in K_n$, so $s[t_0/y] \in K_n^1 \setminus K_n$. In general, if there is a term t_m of rank m, then $s[t_m/y]$ is a term of rank $m+1$. This proves that $K_n^m \subsetneq K_n^{m+1}$ and, in general, that $K_n^m[\bar{x}] \subsetneq K_n^{m+1}[\bar{x}]$.

2. By definition, the head variable y_i of any term in K_n must be in $\bar{y}$. The term $\lambda z_1 \dots z_n.y_1 z_1 \dots z_n$ is in $K_n[\bar{y}] \setminus H_n[\bar{y}]$, where $y_1 \in \bar{y}$: this implies that $t \equiv \lambda \bar{y}.y_1 \dots y_n(\lambda z_1 \dots z_n.y_1 z_1 \dots z_n) \in K_n \setminus H_n$.

□

The following syntactical lemmas are necessary to understand the structure of hereditarily n-ary terms.

Lemma 4.2.7. *If $\bar{y}$ is a sequence of n distinct variables, $s \in T_n[\bar{x}, \bar{y}]$ and $\bar{t} = t_1, \dots, t_n \in T_n[\bar{x}]$, then $s[\bar{t}/\bar{y}] \in T_n[\bar{x}]$.*

Proof. Let $m = \max\{rk(s), rk(t_1) \dots, rk(t_n)\}$. Then $s \in K_n^m[\bar{x}, \bar{y}]$ and $t_i \in K_n^m[\bar{x}]$. By Definition 4.2.3 we obtain $s[\bar{t}/\bar{y}] \in K_n^{m+1}[\bar{x}]$. □

Lemma 4.2.8. *Let t be a λ-term and n, m natural numbers with $n > 0$. Then $t \in K_n^m[\bar{x}]$ if, and only if, there exist*

- *sequences $\bar{z}^i$ $(i = 1, \dots, m)$ of distinct variables,*
- *$s \in K_n^0[\bar{x}, \bar{z}^1, \dots, \bar{z}^m]$,*
- *sequences $\bar{t^i}$ $(i = 1, \dots, m)$ of terms $\bar{t^i} = t_1^i, \dots, t_n^i \in K_n^{m-i}[\bar{x}, \bar{z}^1, \dots, \bar{z}^{i-1}]$*

such that $t \equiv s[\overline{t^m}/\overline{z^m}] \cdots [\overline{t^1}/\overline{z^1}]$.

Proof. The proof is by induction on the index m of $K_n^m[\bar{x}]$.

If $m = 0$, there is nothing to prove.

Let $t \in K_n^{m+1}[\bar{x}]$. By definition of t there exist $s \in K_n^m[\bar{x}, \bar{y}]$ and $\bar{u} = u_1, \dots, u_n \in K_n^m[\bar{x}]$ such that $t = s[\overline{u}/\overline{y}]$, where $\bar{y}$ is a sequence of n distinct variables.

By applying the induction hypothesis to s, there exist m sequences $\bar{z}^i$ of n distinct variables, $s' \in K_n^0[\bar{x}, \bar{y}, \bar{z}^1, \dots, \bar{z}^m]$ and m sequences $\bar{t^i}$ of terms $\bar{t^i} = t_1^i, \dots, t_n^i \in K_n^{m-i}[\bar{x}, \bar{y}, \bar{z}^1, \dots, \bar{z}^{i-1}]$ such that $s \equiv s'[\overline{t^m}/\overline{z^m}] \cdots [\overline{t^1}/\overline{z^1}]$.

This implies that $t \equiv s'[\overline{t^m}/\overline{z^m}] \cdots [\overline{t^1}/\overline{z^1}][\overline{u}/\overline{y}]$. So t is obtained by choosing

- $s' \in K_n^0[\bar{x}, \bar{y}, \bar{z}^1, \dots, \bar{z}^m]$;
- $m + 1$ sequences of n distinct variables $\bar{y}, \bar{z}^1, \dots, \bar{z}^m$;
- $m + 1$ sequences of terms $\bar{u}, \bar{t}^1, \dots, \bar{t}^m$, which satisfy $\bar{u} = u_1, \dots, u_n \in K_n^m[\bar{x}]$ and $\bar{t^i} = t_1^i, \dots, t_n^i \in K_n^{m-i}[\bar{x}, \bar{y}, \bar{z}^1, \dots, \bar{z}^{i-1}]$.

□

4.2.3 Regular mute terms

The following proposition is the main result of this section.

Proposition 4.2.9. *Let $n \geq 1$. If $s_0, s_1, \ldots, s_n$ are restricted hereditarily n-ary terms, then there exist $r_1, \ldots, r_n \in H_n$ and $1 \leq i \leq n$ such that:*

$$s_0 s_1 \ldots s_n \to_h^n s_i r_1 \ldots r_n.$$

In general, given $s_0, s_1, \ldots, s_n \in T_n$, there exist $r_0, r_1, \ldots, r_n \in T_n$ such that:

$$s_0 s_1 \ldots s_n \to_h^n r_0 r_1 \ldots r_n.$$

Proof. Let $s_0, \ldots, s_n \in H_n$. Since $s_0 \in H_n$, then $s_0 \equiv \lambda y_1 \ldots y_n . y_i t_1 \ldots t_n$ with $t_1, \ldots, t_n \in H_n[y_1, \ldots, y]$. Hence $s_0 s_1 \ldots s_n \to_h^n s_i t_1[\bar{s}/\bar{y}] \ldots t_n[\bar{s}/\bar{y}]$, where $\bar{y} \equiv y_1 \ldots y_n$ and $\bar{s} \equiv s_1 \ldots s_n$. By Lemma 4.2.5 the term $t_i[\bar{s}/\bar{y}] \in H_n$, and we are done by defining $r_i \equiv t_i[\bar{s}/\bar{y}]$.

The second part is proved by induction on the rank of s_0.
$rk(s_0) = 0$.
Since $s_0 \in K_n$, then by Lemma 4.2.6.1 $s_0 \equiv \lambda y_1 \ldots y_n . y_i r_1 \ldots r_n$ with $r_1, \ldots, r_n \in K_n[y_1, \ldots, y_n]$. Hence $s_0 s_1 \ldots s_n \to_h^n s_i r_1[\bar{s}/\bar{y}] \ldots r_n[\bar{s}/\bar{y}]$. By Lemma 4.2.7, we have that $r_i[\bar{s}/\bar{y}] \in T_n$.
$rk(s_0) = m > 0$.
By Lemma 4.2.8 there exists $u \in K_n[\bar{z}^1, \ldots, \bar{z}^m]$ such that $s_0 \equiv u[\bar{t}^m/\bar{z}^m] \ldots [\bar{t}^1/\bar{z}^1]$, for some terms $\bar{t}^i \in K_n^{m-i}[\bar{z}^1, \ldots, \bar{z}^{i-1}]$, for all $1 \leq i \leq m$. The term u cannot be a variable because of the rank of s_0. Then by definition $u \equiv \lambda \bar{y}. a u_1 \ldots u_n$ with $a \in \bar{z}^1 \cup \ldots \bar{z}^m \cup \bar{y}$ and $u_i \in K_n^0[\bar{z}^1, \ldots, \bar{z}^m, \bar{y}]$.
We have now two subcases:

(a) if $a \in \bar{y}$, then $u \equiv \lambda \bar{y}. y_i u_1 \ldots u_n$. So we have that:

$$s_0 = \lambda \bar{y}. y_i (u_1[\bar{t}^m/\bar{z}^m] \ldots [\bar{t}^1/\bar{z}^1]) \ldots (u_n[\bar{t}^m/\bar{z}^m] \ldots [\bar{t}^1/\bar{z}^1])$$

and

$$s_0 s_1 \ldots s_n \to_h^n s_i (u_1[\bar{t}^m/\bar{z}^m] \ldots [\bar{t}^1/\bar{z}^1][\bar{s}/\bar{y}]) \ldots (u_n[\bar{t}^m/\bar{z}^m] \ldots [\bar{t}^1/\bar{z}^1][\bar{s}/\bar{y}]).$$

If $s_1, \ldots, s_n$ are in T_n, then by Lemma 4.2.7 all terms $(u_i[\bar{t}^m/\bar{z}^m] \ldots [\bar{t}^1/\bar{z}^1][\bar{s}/\bar{y}])$ are in T_n.

(b) In the other case, i.e., if $a \in \{\bar{z}^1, \ldots, \bar{z}^m\}$, then we rename $a \equiv a_j^i$, to denote that a_j^i belongs to the j^{th} element of the sequence $\bar{z}^i$. So the explicit form of the term s_0 is:

$$s_0 \equiv \lambda \bar{y}. (a_j^i[\bar{t}^m/\bar{z}^m] \ldots [\bar{t}^1/\bar{z}^1])(u_1[\bar{t}^m/\bar{z}^m] \ldots [\bar{t}^1/\bar{z}^1]) \ldots (u_n[\bar{t}^m/\bar{z}^m] \ldots [\bar{t}^1/\bar{z}^1])$$

Thanks to the fact that $\bar{z}^i$ are all distinct variables, we have that

$$(a_j^i[\bar{t}^m/\bar{z}^m] \ldots [\bar{t}^1/\bar{z}^1]) = (t_j^i[\bar{t}^{i-1}/\bar{z}^{i-1}] \ldots [\bar{t}^1/\bar{z}^1])$$

Now we can explicitly compute the first n steps of head reduction of the term $s_0 \dots s_n$:

$$
\begin{aligned}
& s_0 \dots s_n \equiv \\
& (\lambda \bar{y}.(t^i_j[\bar{t}^m/\bar{z}^m]\dots[\bar{t}^1/\bar{z}^1])(u_1[\bar{t}^m/\bar{z}^m]\dots[\bar{t}^1/\bar{z}^1])\dots(u_n[\bar{t}^m/\bar{z}^m]\dots[\bar{t}^1/\bar{z}^1]))s_1\dots s_n \to^n_h \\
& (t^i_j[\bar{t}^m/\bar{z}^m]\dots[\bar{t}^1/\bar{z}^1][\bar{s}/\bar{y}])(u_1[\bar{t}^m/\bar{z}^m]\dots[\bar{t}^1/\bar{z}^1][\bar{s}/\bar{y}])\dots(u_n[\bar{t}^m/\bar{z}^m]\dots[\bar{t}^1/\bar{z}^1][\bar{s}/\bar{y}]) \equiv \\
& (t^i_j[\bar{t}^{i-1}/\bar{z}^{i-1}]\dots[\bar{t}^1/\bar{z}^1])(u_1[\bar{t}^m/\bar{z}^m]\dots[\bar{t}^1/\bar{z}^1][\bar{s}/\bar{y}])\dots(u_n[\bar{t}^m/\bar{z}^m]\dots[\bar{t}^1/\bar{z}^1][\bar{s}/\bar{y}])
\end{aligned}
$$

By Lemma 4.2.7, $(t^i_j[\bar{t}^{i-1}/\bar{z}^{i-1}]\dots[\bar{t}^1/\bar{z}^1])$ and all $(u_i[\bar{t}^m/\bar{z}^m]\dots[\bar{t}^1/\bar{z}^1][\bar{s}/\bar{y}])$ belong to K^m_n.

□

Theorem 4.2.10. *For all $s_0, \dots, s_n \in T_n$, the term $s_0 s_1 \dots s_n$ is mute.*

Proof. By Proposition 4.2.9 there exists an infinite path of head reductions

$$s_0 \dots s_n \to^n_h s^1_0 s^1_1 \dots s^1_n \to^n_h s^2_0 s^2_1 \dots s^2_n \to^n_h \dots s^k_0 s^k_1 \dots s^k_n \to^n_h \dots$$

that has an infinite number of terms with a top redex. By [10, theorem 2.1] $s_0 \dots s_n$ is a mute term. □

Thanks to this theorem, we can define two new classes of mute terms.

Definition 4.2.11. *A term $s_0 s_1 \dots s_n$ where all s_i are in H_n is called **restricted n-regular mute term**; a term $s_0 s_1 \dots s_n$ where all $s_i \in T_n$ is called **n-regular mute term**.*

$\mathcal{M}_n$ denotes the set of all restricted n-regular mute terms; $\mathcal{G}_n$ denotes the set of all n-regular mute terms.

4.2.4 Examples

Some unary and binary regular mute terms:

- $\Omega \equiv (\lambda x.xx)(\lambda x.xx) \in \mathcal{M}_1$
- $(\lambda x.x(\lambda y.yx)(\lambda x.xx) \in \mathcal{M}_1$
- $AAA \in \mathcal{M}_2$, where $A := \lambda xy.x(\lambda zt.tzx)y$.
- $BB \in \mathcal{G}_2$, where $B \equiv \lambda y.y(\lambda x.yx)$.

We give now an example of a a mute term that is not regular. Let $U \equiv \lambda xy.y(xxy)$ and consider the term UUI. We have that

$$
\begin{array}{ll}
UUI & \to^2_h \\
I(UUI) & \to^1_h \\
UUI &
\end{array}
$$

so the term UUI is mute. On the other hand, U is not a hereditarily n-ary term because in it two lambda abstractions bind a term which is an application of only two terms, y and xxy: according to the definition there should be three.

The next two lemmas are not necessary for the forthcoming of the chapter. We proved them when attempting to generalize the graph-easiness property proved for restricted regular mutes. Nonetheless, we believe they can be useful in further research.

Lemma 4.2.12. *Let $t \in T_n[\bar{x}]$. Then its form is $\lambda\bar{y}^1.(\lambda\bar{y}^2.(\ldots(\lambda\bar{y}^m.v\bar{t}^m)\ldots)\bar{t}^2)\bar{t}^1$ where $m \geq 0$, each $\bar{y}^i$ is a sequence of exactly n fresh and distinct variablesand v is a variable among $\bar{x}$ or $\bar{y}^m$. We denote by t^i_j the j^{th} term of the sequence $\bar{t}^i$: then we have that $FV(t^i_j) \subseteq \bar{y}^i \cup \bar{x}$, for all $1 \leq i \leq m$ and $1 \leq j \leq n$.*

Proof. Let $t \in K^m_n[\bar{x}]$. Proof is by induction on m.

- If $t \in K^0_n[\bar{x}]$, then conclusion trivially follows by definition of $K^0_n[\bar{x}]$.
- Let $t \in K^{m+1}_n[\bar{x}]$. By definition, $t \equiv s[\bar{u}/\bar{z}]$, with $s \in K^m_n[\bar{x}, \bar{z}]$ and $\bar{u} \subseteq K^m_n[\bar{x}]$. So we can use the inductive hypothesis on s. If the first variable in its body is is bounded by the lambda's immediately preceding it or is among the $\bar{x}$'s, conclusion immediately follows. Otherwise, i.e. if it is among the $\bar{z}$'s, say z', it is replaced by one of the $\bar{u}$. By applying the inductive hypothesis on the specific u_i that substitutes z' we get the result.

Now we prove the second part of the lemma.

- If $t \in K^0_n[\bar{x}]$, then it is of the form $\lambda\bar{y}.vt_1 \ldots t_n$, with $v \in \bar{x}\cup\bar{y}$ and $t_j \in K^0_n[\bar{x}, \bar{y}]$, so conclusion trivially holds.
- Let $t \in K^{m+1}_n[\bar{x}]$. By definition, $t \equiv s[\bar{u}/\bar{z}]$, where $s \in K^m_n[\bar{x}, \bar{z}]$ and $\bar{u} \subseteq K^m_n[\bar{x}]$. By the first result of the lemma and inductive hypothesis we know that $s \equiv \lambda\bar{y}^1.(\lambda\bar{y}^2.(\ldots(\lambda\bar{y}^m.v\bar{t}^m)\ldots)\bar{t}^2)\bar{t}^1$, with $v \in \bar{y}^m\cup\bar{x}\cup\bar{z}$ and $\mathrm{FV}(\bar{t}^i) \subseteq \bar{y}^i\cup\bar{x}\cup\bar{z}$. Thanks to the fact that $\bar{u} \subseteq K^m_n[\bar{x}]$, the conclusion follows.

□

Corollary 4.2.13. *Let s_0 be an hereditarily n-ary term of the form*

$$\lambda\bar{y}^1.(\lambda\bar{y}^2.(\ldots(\lambda\bar{y}^n.y^n_i\bar{t}^n)\ldots)\bar{t}^2)\bar{t}^1.$$

Then in the head reduction path of a regular mute $s_0s_1 \ldots s_n$, we have that after $|\bar{y}^1| + \cdots + |\bar{y}^{n-1}| = n(n-1)$ steps we get a term of the form $(\lambda\bar{y}.y_it_1 \ldots t_p)s_1 \ldots s_n$.

4.2.5 Decidability of sets $\mathcal{M}_n$ and $\mathcal{G}_n$

In the following proofs, $l(t)$ denotes the length of the term t.

Proposition 4.2.14. *For all $n \geq 1$, sets $\mathcal{G}_n$ and $\mathcal{M}_n$ are recursive.*

Proof. If t is a n-regular mute term, then $t \equiv t_0 \dots t_n$ for some $t_0, \dots, t_n \in T_n$. Then we have that $\mathcal{G}_n$ is recursive if and only if T_n is such. Similarly, $\mathcal{M}_n$ is recursive iff H_n is such. Without loss of generality, we now prove that sets $T_n[\bar{x}]$ and $H_n[\bar{x}]$ are recursive for every finite sequence $\bar{x}$ of distinct variables.

Claim 4.2.15. *For every $m \geq 0$, $n \geq 1$ and every sequence $\bar{x}$ of distinct variables, sets $H_n[\bar{x}]$ and $K_n^m[\bar{x}]$ are recursive.*

Proof. The proof is by induction on m.

- $m = 0$. Let t be an arbitrary λ-term. We check the relations $t \in K_n[\bar{x}]$ and $t \in H_n[\bar{x}]$ by induction on the length of t.
  - If t is a variable, then $t \in K_n[\bar{x}]$ iff $t \in \bar{x}$. The same holds for $H_n[\bar{x}]$
  - If $t \equiv \lambda\bar{y}.zt_1 \dots t_n$, where z is a variable, $\bar{y}$ is a sequence of distinct variables and t_i are arbitrary terms, then it is recursive to check if $z \in \bar{y} \cup \bar{x}$ and, thanks to inductive hypothesis, whether all t_i are in $K_n[\bar{x}]$ or not. Proof for $H_n[\bar{x}]$ is similar: the only difference is that z must be in $\bar{y}$.
  - $t \notin K_n[\bar{x}]$ or $t \notin H_n[\bar{x}]$, otherwise.

  This ends the proof for $\mathcal{M}_n$.

- Assume by induction hypothesis that $K_n^m[\bar{x}]$ is recursive for every sequence $\bar{x}$ of distinct variables. We prove that $K_n^{m+1}[\bar{x}]$ is recursive. By Definition 4.2.3, $t \in K_n^{m+1}[\bar{x}]$ if and only if $t \equiv u_0[\bar{u}/\bar{y}]$, for some $u_0 \in K_n^m[\bar{x}, \bar{y}]$ and $\bar{u} \equiv u_1, \dots, u_n \in K_n^m[\bar{x}]$. We have that $l(u_i) \leq l(t)$ for every $0 \leq i \leq n$. Consider $l(t)$ distinct new variables $\bar{z} \equiv z_1, \dots, z_{l(t)}$. Since $K_n^m[\bar{x}, \bar{z}]$ and $K_n^m[\bar{x}]$ are recursive and $\{u : l(u) \leq l(t)\}$ is finite, then after a finite time we get the sets $U_0 = \{u : l(u) \leq l(t) \wedge u \in K_n^m[\bar{x}, \bar{z}]\}$ and $U_1 = \{u : l(u) \leq l(t) \wedge u \in K_n^m[\bar{x}]\}$. We have that $t \in K_n^{m+1}[\bar{x}]$ if and only if $t \equiv u_0[\bar{u}/\bar{y}]$, for some $u_0 \in U_0$ and $\bar{u} \equiv u_1, \dots, u_n \in U_1$. This can be decided in a finite time.

□

Claim 4.2.16. $t \in T_n[\bar{x}] \Rightarrow \mathrm{rk}(t) \leq l(t)$.

Proof. The proof is by induction on the length of t. Let $m + 1 = \mathrm{rk}(t)$ in this proof. Since $t \in K_n^{m+1}[\bar{x}]$, then there exist $u_0 \in K_n^m[\bar{x}, \bar{y}]$ and $\bar{u} \equiv u_1, \dots, u_n \in K_n^m[\bar{x}]$ such that $t \equiv u_0[\bar{u}/\bar{y}]$. Without loss of generality, we assume that all variables in $\bar{y}$ occur free in u_0. The following statements hold:

- by definition of $\mathrm{rk}(t)$ there exists $0 \leq j \leq n$ such that $\mathrm{rk}(u_j) = m$.
- for all $0 \leq i \leq n$, $l(u_i) < l(t)$. Suppose that $l(u_i) = l(t)$ for some i:
  - if $1 \leq i \leq n$, then $l(u_i) = l(t)$ implies $u_i \equiv t$.
  - if $i = 0$, then $l(u_0) = l(t)$ implies $u_0 \equiv t$ up to renaming of free variables in $\bar{x}$.

  In both cases, we have that $\mathrm{rk}(t) = \mathrm{rk}(u_i)$ for some $0 \leq i \leq n$, contradiction.
- for all $0 \leq i \leq n$, $\mathrm{rk}(u_i) \leq l(u_i)$, by induction hypothesis on $l(u_i)$.

Hence $\mathrm{rk}(t) = \mathrm{rk}(u_j) + 1 \leq l(u_j) + 1 \leq l(t)$.
In conclusion, $t \in T_n[\bar{x}]$ iff $t \in \bigcup_{0 \leq i \leq l(t)} K_n^i[\bar{x}]$, which is a recursive set since it is a finite union of recursive sets.

□

□

4.3 Graph-easiness of restricted n-regular mute terms

In this section we show that, given a closed λ-term t and a natural number n, there exists a graph model which equates all the restricted n-regular mute terms. The proof requires some technical lemmas and then an application of the forcing for graph models as described in [13].

4.3.1 Some useful lemmas

Lemma 4.3.2 below generalizes Proposition 2.4.6 obtained by Baeten and Boerboom in [4].

Lemma 4.3.1. *Let (D, p) be a graph model, ρ be an environment, $\beta \in D$, and $\bar{\beta} = \beta, \beta, \ldots, \beta$ (n-times). If $\beta = \bar{\beta} \to \alpha$, $t \in H_n[\bar{x}]$ and $\beta \in \rho(x_i)$ $(i = 1, \ldots, k)$ then $\beta \in |t|_\rho^p$.*

Proof. The proof is by induction over the complexity of t as hereditarily n-ary λ-term. If $t \equiv x_i$ then the conclusion is trivial because $\beta \in \rho(x_i)$. Otherwise, there exists $\bar{u} \equiv u_1, \ldots, u_n \in H_n[\bar{x}, \bar{y}]$ such that $t = \lambda \bar{y}.y_i \bar{u}$.

$$\beta = \bar{\beta} \to \alpha \in |\lambda \bar{y}.y_i \bar{u}|_\rho^p \;\Leftrightarrow\; \alpha \in |y_i \bar{u}|^p_{\rho[\bar{y}:=\bar{\beta}]}.$$

Since $\beta \in \rho[\bar{y} := \bar{\beta}](y_i)$ for every $i = 1, \ldots, n$, then by induction hypothesis $\beta \in |u_i|^p_{\rho[\bar{y}:=\bar{\beta}]}$ for every $i = 1, \ldots, n$. It follows that $\alpha \in |y_i \bar{u}|^p_{\rho[\bar{y}:=\bar{\beta}]}$ and we get the conclusion. □

Lemma 4.3.2. *Let (D, p) be a graph model, $t \in \mathcal{M}_n$ and $\gamma \in |t|^p$. Then there exist a sequence $\beta_i \equiv a^i_1 \to \cdots \to a^i_n \to \gamma$ $(i \in \omega)$ of elements of D and a sequence d_i $(i \in \omega)$ of natural numbers $\leq n$ such that $\beta_{i+1} \in a^i_{d_i}$.*

Proof. Let $t \equiv s_0 \dots s_n$. By Proposition 4.2.9 there exists an infinite sequence of mute terms such that

$$s^0_0 s^0_1 \dots s^0_n \ \to^n_\beta \ s^1_0 s^1_1 \dots s^1_n \ \to^n_\beta \ \dots \ \to^n_\beta \ s^k_0 s^k_1 \dots s^k_n \ \to^n_\beta \dots$$

and $s^k_0 \equiv s^{k-1}_{d_{k-1}}$ for some $1 \leq d_{k-1} \leq n$. The number d_{k-1} is the order of the head variable of the term s^{k-1}_0. By $\gamma \in |s^0_0 s^0_1 \dots s^0_n|^p$ there exists $a^0_1 \to \cdots \to a^0_n \to \gamma \in |s^0_0|^p$ such that $a^0_i \subseteq |s^0_i|^p$. We define

$$\beta_0 = a^0_1 \to \cdots \to a^0_n \to \gamma.$$

Assume $\beta_k = a^k_1 \to \cdots \to a^k_n \to \gamma \in |s^k_0|^p$ and $a^k_j \subseteq |s^k_j|^p$ for every $j \leq n$. Since $s^k_0 = \lambda\bar{y}.y_{d_k} u_1 \dots u_n$ for some terms u_i and $\beta_k \in |s^k_0|^p$, then we have

$$\gamma \in a^k_{d_k} |u_1[\bar{a}/\bar{y}]|^p \dots |u_n[\bar{a}/\bar{y}]|^p,$$

where $\bar{a} = a^k_1, \dots, a^k_n$. It follows that there exists

$$\beta_{k+1} = a^{k+1}_1 \to \cdots \to a^{k+1}_n \to \gamma \in a^k_{d_k} \tag{4.1}$$

such that $a^{k+1}_j \subseteq |u_j[\bar{a}/\bar{y}]|^p$. We have to prove that $\beta_{k+1} \in |s^{k+1}_0|^p$ and $a^{k+1}_j \subseteq |s^{k+1}_j|^p$ for every $j \leq n$. By applying the induction hypothesis and (4.1) we get $\beta_{k+1} \in a^k_{d_k} \subseteq |s^k_{d_k}|^p = |s^{k+1}_0|^p$. The other relation can be obtained as follows, by defining $\bar{s}^k = s^k_1, \dots, s^k_n$: $a^{k+1}_j \subseteq |u_j[\bar{a}/\bar{y}]|^p \subseteq |u_j[|\bar{s}^k|^p/\bar{y}]|^p = |s^{k+1}_j|^p$. $\square$

4.3.2 Forcing at work

In this section we apply the forcing technique to build, for any $t \in \Lambda^0$, the desired graph model equating it to all elements of $\mathcal{M}_n$. The notion of forcing (Definition 4.3.10) will be introduced when necessary, after Theorem 4.3.4.

Let $\mathcal{I}(D)$ be the cpo of partial injection from $D^* \times D$ into D. If $p \in \mathcal{I}(D)$ then the *universe* $\mathrm{Un}(p)$ of p is defined as follows:

$$\mathrm{Un}(p) = \bigcup_{(a,\alpha)\in dom(p)} (a \cup \{\alpha, p(a, \alpha)\}).$$

If p is finite, then the universe of p is also finite.

Definition 4.3.3. *Let $p \in \mathcal{I}(D)$ be finite, $\alpha \in D$ and $\bar{\epsilon} \equiv \epsilon_1, \dots, \epsilon_k \in D \setminus \mathrm{Un}(p)$. Then $p_{\bar{\epsilon},\alpha}$ is the extension of p such that $\epsilon_{k+1} = \alpha$ and $\epsilon_j = \epsilon_1 \to \epsilon_{j+1}$ $(j = 1, \dots, k)$.*

Notice that

$$\epsilon_1 = \underbrace{\epsilon_1 \to \epsilon_1 \to \cdots \to \epsilon_1}_{k\text{-times}} \to \alpha$$

and $p_{\bar{\epsilon},\alpha}$ is also finite.

The next theorem is the main technical tool for proving the easiness of the set of restricted e-regular mute terms. It generalizes [13, theorem. 11].

Theorem 4.3.4. *Let $F : \mathcal{I}(D) \to \mathcal{P}(D)$ be a weakly continuous function and let $e \in \mathbb{N}$. Then there exists a total $p' : D^* \times D \to D$ such that $(D, p') \models |t|^{p'} = F(p')$ for all restricted e-regular mute terms t.*

Proof. We are going to build an increasing sequence of finite injective maps $p_n : D^* \times D \rightharpoonup D$, starting from $p_0 = \emptyset$, and a sequence of elements $\alpha_n \in D \cup \{*\}$, where $*$ is a new element, such that: $p' =_{\text{def}} \cup p_n$ is a total injection, and $(D, p') \models |t|^{p'} = F(p') = \{\alpha_n : n \in \omega\} \cap D$, for all $t \in \mathcal{M}_e$.

We fix an enumeration of D and an enumeration of $D^* \times D$.

We start from $p_0 = \emptyset$.

Assume that $p_n : D^* \times D \rightharpoonup D$ and $\alpha_0, \ldots, \alpha_{n-1}$ have been built. We let

- $\alpha_n =$ First element of $F(p_n) \setminus \{\alpha_0, \ldots, \alpha_{n-1}\}$ in the enumeration of D, if this set is non-empty, and $\alpha_n = *$ otherwise;

- $(b_n, \delta_n) =$ First element in $(D^* \times D) \setminus dom(p_n)$;

- $\gamma_n =$ First element in $D \setminus (\text{Un}(p_n) \cup b_n \cup \{\delta_n\} \cup \{\alpha_0, \ldots, \alpha_{n-1}, \alpha_n\})$.

We define a new finite injection r as follows:

$$r(\beta) = \begin{cases} p_n(\beta) & \text{if } \beta \in dom(p_n) \\ \gamma_n & \text{if } \beta = (b_n, \delta_n) \end{cases}$$

Case 1: $\alpha_n = *$. We let $p_{n+1} = r$.

Case 2: $\alpha_n \in D$.

We define $p_{n+1} = r_{\bar{\epsilon}^n, \alpha_n}$ (see Definition 4.3.3), where

$$\bar{\epsilon}^n = \epsilon_1^n, \ldots, \epsilon_e^n \in D \setminus (\text{Un}(r) \cup \{\alpha_n\})$$

are the first e distinct elements of $D \setminus (\text{Un}(r) \cup \{\alpha_n\})$.

It is clear that p_n is a strictly increasing sequence of well-defined finite injective maps and that $p' = \cup p_n$ is total.

It is also clear that each p_n (and p') is partitioned into two disjoint sets: $p_n = p_n^1 \cup p_n^2$, where $p_n^1 = \{b_i \to \delta_i = \gamma_i : 1 \leq i \leq n-1\}$ is called the gamma part of p_n and $p_n^2 = p_n \setminus p_n^1$ is called the epsilon part.

For every $\gamma \in D$, we define

$$deg(\gamma) = \begin{cases} 0 & \text{if } \gamma \notin rg(p') \\ min\{n : \gamma \in rg(p_n)\} & \text{if } \gamma \in rg(p') \end{cases}$$

Moreover, $deg(c) = max\{deg(x) : x \in c\}$ for every $c \subseteq_{\text{fin}} D$.

The following claims easily derive from the construction of p'.

Claim 4.3.5. $\forall n' > n,\ (rg(p_{n'}) \setminus rg(p_n)) \cap \text{Un}(p_n) = \emptyset.$

Proof. Let $\beta \in rg(p_{n'}) \setminus rg(p_n)$ and $k = deg(\beta)$. Then $n < k \leq n'$. Since $\beta \in rg(p_k) \setminus rg(p_{k-1})$ and β can be either γ_{k-1} or one of the ϵ_j^{k-1}, then β is not an element of $\text{Un}(p_{k-1})$ by construction of p_k. Since $\text{Un}(p_n) \subseteq \text{Un}(p_{k-1})$, we get the conclusion. □

Claim 4.3.6. *If $deg(a \to \alpha) = n$ and $\alpha \notin rg(p_n)$, then $\alpha \notin rg(p')$.*

Proof. If $\alpha \in rg(p')$ then $\alpha \in rg(p_j)$ for some j. By $\alpha \notin rg(p_n)$, it must be $n < j$. We also have that $p'(a, \alpha) = p_n(a, \alpha)$, for $deg(p'(a, \alpha)) = n$. Thus $(a, \alpha) \in dom(p_n)$ and $\alpha \in \text{Un}(p_n)$. By Claim 4.3.5 and $\alpha \in rg(p_j) \setminus rg(p_n)$, we get a contradiction. □

Claim 4.3.7. *(i) $deg(a \to \alpha) \geq deg(a \cup \{\alpha\})$.*
(ii) If $a \to \alpha$ is in the gamma part of p', then $deg(a \to \alpha) > deg(a \cup \{\alpha\})$.

Proof. Let $deg(a \to \alpha) = n$. Thus, $p'(a, \alpha) = p_n(a, \alpha)$ and $a \cup \{\alpha\} \subseteq \text{Un}(p_n)$.

(i) If $n < deg(\alpha) = j$, then $\alpha \in rg(p_j) \setminus rg(p_n)$, that contradicts Claim 4.3.6. It follows that $deg(\alpha) \leq n$.

If $n < deg(a) = j$, then there exists $\theta \in a$ such that $deg(\theta) = j > n$, so that $\theta \in rg(p_j) \setminus rg(p_n)$. By Claim 4.3.5 $\theta \notin \text{Un}(p_n)$. This contradicts $a \subseteq \text{Un}(p_n)$.

(ii) By (i) it is sufficient to show that $deg(a \cup \{\alpha\}) \neq n$. By hypothesis $a = b_{n-1}$, $\alpha = \delta_{n-1}$ and $p_n(a, \alpha) = \gamma_{n-1}$. Then by construction $deg(\gamma_{n-1}) = n$. By definition of p_n, α is different from γ_{n-1} and from any ϵ_j^{n-1}. So it cannot be in $rg(p_n) \setminus rg(p_{n-1})$. The same reasoning applies to $a = b_{n-1}$. □

Claim 4.3.8. *If $\alpha_n \in rg(p')$, then $deg(\alpha_n) \leq n$.*

Proof. By construction of p' we have that $\alpha_n \in \text{Un}(p_{n+1})$ and $\alpha_n \notin rg(p_{n+1}) \setminus rg(p_n)$. Then $deg(\alpha_n) \neq n+1$. If $deg(\alpha_n) = j > n+1$, then $\alpha_n \in rg(p_j) \setminus rg(p_{n+1})$. This togehter with $\alpha_n \in \text{Un}(p_{n+1})$ contradicts Claim 4.3.5. □

Claim 4.3.9. *The total map p' contains no cycle $\beta = c_1 \to c_2 \to \ldots c_m \to \beta$.*

Proof. Consider a minimal cycle $\beta_i = c_i \to \beta_{i+1}$ $(1 \leq i \leq m-1)$ and $\beta_m = c_m \to \beta_1$. By Claim 4.3.7 we have $deg(\beta_1) \geq deg(\beta_2) \geq \cdots \geq deg(\beta_m) \geq deg(\beta_1)$. Let us set this common degree equal to $k+1$. If $\beta_1 = \gamma_k = b_k \to_{p_{k+1}} \delta_k$ then $\delta_k = \beta_2$ has degree $k+1$. This is not possible by Claim 4.3.7(ii). If $\beta_1 = \epsilon_j^k$ then $\epsilon_j^k = c_1 \to c_2 \to \ldots c_m \to \epsilon_j^k$. From this it follows that either α_k has degree $k+1$ (contradicting

Claim 4.3.8) or $\epsilon_j^k = \epsilon_{j-l}^k$ (contradicting that the epsilon elements are distinct) or $\epsilon_j^k = \alpha_k$ (contradicting the definition of epsilon elements). This concludes the proof of the claim. $\square$

Let $X = \{\alpha_n : n \in \omega\} \cap D$. We now show that $(D, p') \models X = F(p')$.

- $X \subseteq F(p')$: it follows from the definition of α_n and from the fact that $F(p_n) \subseteq F(p')$.
- $F(p') \subseteq X$: suppose $\gamma \in F(p')$; since F is weakly continuous, $\gamma \in F(p_i)$ for some i (and for all the larger ones). If $\gamma \notin X$ then, for all $n \geq i$, $\alpha_n \neq *$ has smaller rank than γ in the enumeration of D, contradicting the fact that there is only a finite number of such elements.

Let $t \equiv s_0 s_1 \dots s_e \in \mathcal{M}_e$. Now we show that $(D, p') \models X = |t|^{p'}$.

- $X \subseteq |t|^{p'}$: Let $\alpha_n \neq *$. The condition $(D, p') \models \alpha_n \in |t|^{p'}$ follows immediately from Lemma 4.3.1 and the fact that

$$\epsilon_1^n = \epsilon_1^n \to \epsilon_1^n \to \cdots \to \epsilon_1^n \to \alpha_n \quad (e\text{-times}).$$

- $|t|^{p'} \subseteq X$: Assume that $\gamma \in |t|^{p'}$. Then by Lemma 13 there exists a sequence $\beta_j \equiv a_1^j \to \cdots \to a_e^j \to \gamma$ $(j \in \omega)$ of elements of D and a sequence d_j $(j \in \omega)$ of natural numbers $\leq e$ satisfying the property $\beta_{j+1} \in a_{d_j}^j$.

  By Claim 17 and by $\beta_{j+1} \in a_{d_j}^j$ the sequence $deg(\beta_j)$ is an infinite decreasing sequence of natural numbers. Then there exist i and n such that $deg(\beta_k) = n+1$ for all $k \geq i$.

  There are (at most) $e+1$ elements having degree $n+1$, namely

  $\gamma_n = b_n \to \delta_n$

  $\epsilon_1^n = \epsilon_1^n \to \cdots \to \epsilon_1^n \to \alpha_n$ (e-times)

  $\epsilon_2^n = \epsilon_1^n \to \cdots \to \epsilon_1^n \to \alpha_n$ ($e-1$-times)

  $\dots$

  $\epsilon_e^n = \epsilon_1^n \to \alpha_n$.

  Since $deg(\beta_i) = n+1$, then β_i is one of the element listed above, too. We have many possibilities:

  (1): $\beta_i \equiv \gamma_n = b_n \to \delta_n$ is not possible. In fact, by the definition of β_i we derive that $b_n \equiv a_1^i$ and $\delta_n \equiv a_2^i \to \cdots \to a_e^i \to \gamma$. By Claim 4.3.7(ii) $deg(b_n)$ and $deg(\delta_n)$ are strictly less than $n+1$, so that $deg(a_j^i) < n+1$ for every $1 \leq j \leq e$. From $\beta_{i+1} \in a_{d_i}^i$ we get the contradiction $deg(\beta_{i+1}) < n+1$.

  Then we must have that $\beta_i \equiv \epsilon_r^n$ for some r.

(2) $\beta_i \equiv \epsilon_r^n = \epsilon_1^n \to \cdots \to \epsilon_1^n \to \alpha_n$ ($e - r + 1$-times). By the definition of β_i we derive that $a_j^i = \{\epsilon_1^n\}$ for every $1 \le j \le e-r+1$ and $\alpha_n \equiv a_{e-r+2}^i \to \cdots \to a_e^i \to \gamma$. But by Claim 4.3.8 we have that $deg(\alpha_n) < n+1$. This implies that $\beta_{i+1} = \epsilon_1^n$ and then that $\gamma = \alpha_n$.

Since the previous points implies that $(D, p') \models |t|^{p'} = F(p')$, this concludes the proof of Theorem 4.3.4. $\square$

Now we introduce forcing for graph models. This technique allow us to build a weakly continuous function $F_{t,\rho}$ suitable for Theorem 4.3.4 by applying Lemma 4.3.11.

Definition 4.3.10. Forcing for graph models. *For a term t, a D-environment ρ, a partial pair (D, q) and $\alpha \in D$, the abbreviation $q \Vdash_\rho \alpha \in t$ means that for all total injections $p \supseteq q$ we have that $(D, p) \models \alpha \in |t|_\rho^p$. Furthermore $q \Vdash_\rho Y \subseteq t$ means that $q \Vdash_\rho \alpha \in t$ for all $\alpha \in Y$.*

If t is closed then we drop ρ. Then we write $q \Vdash \alpha \in t$ for $q \Vdash_\rho \alpha \in t$.
Thus, for p is total, $p \Vdash \alpha \in t$ if and only if $\alpha \in |t|^p$.

Lemma 4.3.11. *For every term t and environment ρ the function $F_{t,\rho} : \mathcal{I}(D) \to \mathcal{P}(D)$ defined by $F_{t,\rho}(q) = \{\, \alpha \in D : q \Vdash_\rho \alpha \in t\}$ is weakly continuous, and we have $F_{t,\rho}(p) = |t|_\rho^p$ for each total p.*

Proof. The proof of the weak continuity of $F_{t,\rho}$ is a straightforward induction on the complexity of t. Let $p \in Q$ be total. We have to show that $F_{t,\rho}(p) = \bigcup_{q \subseteq_{\mathrm{fin}} p} F_{t,\rho}(q) = |t|_\rho^p$.

If t is a variable x then $F_{x,\rho}(q) = \{\, \alpha \in D : q \Vdash \alpha \in \rho(x)\}$ is the constant function with value $\rho(x)$.

If $t = us$ and $\alpha \in |t|_\rho^p$, then there exists $a \subseteq |s|_\rho^p$ such that $p(a, \alpha) \in |u|_\rho^p$. Choose such an a and let $\gamma = p(a, \alpha)$. By induction hypothesis there is a finite $q \subseteq p$ such that $q \Vdash_\rho a \subseteq s$ and a finite $r \subseteq p$ such that $r \Vdash_\rho \gamma \in u$; then it is clear that $q \cup r \cup \{((a, \alpha), \gamma)\} \Vdash \alpha \in t$.

If $t = \lambda x.u$ and $\alpha \in |t|_\rho^p$ then there is a unique pair (b, β) such that $\alpha = p(b, \beta)$ and $\beta \in |u|_{\rho[x:=b]}^p$. By induction hypothesis there is a finite $q \subseteq p$ such that $q \Vdash_{\rho[x:=b]} \beta \in u$; then it is clear that $q \cup \{((b, \beta), \alpha)\} \Vdash_\rho \alpha \in t$. $\square$

Theorem 4.3.12. *Let t be a closed term. Then, for every natural number n there exists a graph model (D, p') such that $(D, p') \models t = u$ for all n-regular mute terms $u \in \mathcal{M}_n$.*

Proof. We take $F_{t,\rho}$ as defined in Lemma 4.3.11, where ρ is any environment. Then we apply Theorem 4.3.4 to get a graph model (D, p') satisfying the condition of the theorem. $\square$

II

Factor algebras for classical logic

5
Introduction

At the beginning of 19^{th} century, Peacock [15] defined algebras whose operations differ from the ordinary operations between numbers. In 1830, De Morgan, in *Trigonometry and Double Algebra*, removed the usual interpretation of the variables as numbers. These were great improvements, but only in 1847 Boole did the decisive step in *The Mathematical Analysis of Logic*: he explicitly states that mathematics can be seen as a purely formal system of symbols and operations on them, whose only need is internal consistency.

This revolutionary idea was one of boosts of the huge development of the algebra in the second half of the 19^{th} century. It was used to grasp the underlying algebraic notions in “concrete”, relevant structures, that could later be used to define abstract algebras. For example, Galois introduced the notion of group after studying the permutations of a set, while the notion of ring generalizes the algebraic structure of the set of integers (Dedekind and Hilbert, [37]).

In 1898 Whitehead, in *A Treatise on Universal Algebra*, introduced for the first time a notion of Universal Algebra, i.e., a general theory for structures with an arbitrary number of finitary operations. Unfortunately, he did not have unifying, relevant results so its book did not have a big influence on the development of algebra. Thirty-seven years later, Garrett Birkhoff, in *On the Structure of Abstract Algebras* provided the explicit foundation of Universal Algebra by introducing the fundamental notions of subalgebra, congruence, variety and by relating it to the theory of lattices. After this work was published, Universal Algebra greatly developed. In ’50s Mal’cev started the characterization of properties of varieties by the existence of certain terms involved in certain identities, called *Mal’cev conditions*.

Another important area of research is the relationship between Universal Algebra and Logic. The algebraic analysis of logic dates to the 17^{th} century with Leibniz (see also Jacob Bernoulli [24]). These were only blurry attempts that did not produce any further development.

In 1847, in *The Mathematical Analysis of Logic* Boole introduced Boolean Algebras, which provide a symbolic formulation of logical problems with equations, to be solved by means of algebraic technique. Some years later De Morgan introduced the logic of relations in the pursuit of generalizing Aristotelian syllogisms. This new field of research was further developed by Peirce [15], that in *On the Algebra of*

Logic also introduced the notion of quantification. In the last part of 19^{th} century, Schröder [24] attempted to develop a general algebraic theory of relations that could be applied to many areas of mathematics.

At the beginning of the 20^{th} century the interest on algebraic methods in logic declined, since the logical methods and notations introduced in *Principia Mathematica* by Russell and Whitehead were accepted by the majority of the mathematical community.

In '40s research in algebraic logic had a new boosts. Tarski introduced, in *On the Calculus of Relations*, a pure and axiomatic equational theory of the calculus of relations. In a joint work with Givant [33] he provided an equational logic of relations in which first-order set theory can be completely expressed .

The other main contribution of Tarski to algebraic logic (in collaboration with Henkin and Monk) are cylindric algebras [63]. Cylindric algebras provide an algebraic theory for studying first-order logic with equality, as Boolean algebras do for propositional logic. Tarski also gave purely algebraic proofs of logical theorems, such as the completeness theorem for first-order logic. Halmos [34] used a similar approach to define Polyadic algebras, which provide an algebraic treatment of first-order predicate logic on an arbitrary signature (so also without equality). Since cylindric and polyadic algebras are complex structures, the research on this field almost completely stopped after the conference on algebraic logic of Budapest (1988).

Algebraic logic is not restricted to classical logic. In 1930 Heyting provided an algebraic semantic to intuitionistic logic by introducing Heyting algebras. In '50s Chang algebraized the infinite-valued Łukasiewitz logic with Multi-Valued (MV-)algebras. Algebraic logic is particularly helpful when dealing with substructural logics. These logics were originally defined as Gentzen-style systems lacking some of the structural rules. The substructural logic is algebraized by a residuated lattice. From a mathematical point of view, it is easier to work with these algebraic structures. Now the theory of substructural logics is mainly developed through algebraic methods rather than logical ones (see [32]).

In the last decades, researchers try to find simple algebraic formalizations of first-order logic:

(i) Manca and Salibra in [50] show that every first-order theory is a particular many-sorted algebra verifying some equational axioms.

(ii) Burris in [22] exhibited a procedure to cast any mathematical problem in the form:

> Is it possible to derive the absurd statement $x = y$ from a particular set of equations?

He used discriminator varieties and Skolemization as main tools.

(iii) Mycielski in [53] proposed a similar translation but without Skolemization.

There are many possible answers to the problem of reducing first-order logic to Universal Algebra, but none of them is completely satisfactory. For example, Tarski's cylindric algebras are too complicated, while in Burris's procedure the length of the equational proofs grows exponentially (this is a consequence of the Skolemization). Nevertheless, research is very active in this area.

In this thesis we give our contribution to Algebraic Logic. We introduce the *factor algebras of a first-order type* generalizing factor algebras, recently introduced in [58]. We show that they are the natural algebraic counterpart of first order structures. Roughly speaking, a relation symbol R of arity n is transformed into an operation symbol $\widehat{R}$ of arity $n+2$ in such a way that $\widehat{R}$ is a decomposition operator in the last two coordinates. In this way, given a first-order type τ, we define a bijective correspondence between non-singleton τ-structures and non-trivial τ-factor algebras: a factor algebra $\mathrm{Fa}(\mathcal{M})$ corresponds to a τ-structure $\mathcal{M}$; and a τ-structure $\mathrm{Str}(\mathbf{A})$ corresponds to a factor algebra $\mathbf{A}$. We extend the correspondence to universal sentences and equations between terms: given a universal τ-sentence ϕ, its translation $\phi^*(y_{\mathrm{f}}, y_{\mathrm{t}})$ is a term in two variables. We prove that these correspondences are semantically meaningful, in the sense that $\mathcal{M} \models \phi$ iff $\mathrm{Fa}(\mathcal{M}) \models \phi^*(y_{\mathrm{f}}, y_{\mathrm{t}}) = y_{\mathrm{t}}$. We start the study of first-order logic through varieties of factor algebras.

In the last part of this thesis, we study propositional logic through factor algebras. Each propositional variable becomes a binary decomposition operator in its algebraic translation. We show that the axioms defining the variety of factor algebras can be used as rules for an algebraic calculus for propositional logic. We show that a propositional formula ϕ is a tautology iff the equation $\phi^* = y_{\mathrm{t}}$ can be proved by the axioms of factor algebras. This methodology gives a new complete calculus for propositional logic. We provide a term rewriting system for the calculus, and we show that it is confluent and terminating.

6 Preliminaries

6.1 First Order Logic

In this section we follow [30].

Definition 6.1.1. *A* first-order type τ *is a set of symbols consisting of logical symbols and parameters:*

- Logical symbols*: the left and right parentheses (,); the negation symbol $\neg$; the conjunction and disjunction symbols $\wedge, \vee$; the conditional symbol $\Rightarrow$; the universal quantifier $\forall$; denumerably many variables $x, y, z, \ldots$.*

- Parameters*: for any $n \geq 1$, some (possibly none) symbols $R, S, \ldots$ of arity n called predicate symbols; for any $n \geq 1$, some (possibly none) symbols $f, g, \ldots$ of arity n called functions symbols; some (possibly none) symbols $c, d, \ldots$ called constants symbols.*

- *The denumerable set of variables is usually denoted by* Var.

Definition 6.1.2. *Let τ be a first-order type.*

- *A* τ-word *is a finite sequence of symbols of τ.*

- *A* τ-term *is a τ-word built inductively according to the following rules:*

  - *all variables and constants are terms.*
  - *If f is a function symbol of arity n and $t_1, \ldots, t_n$ are terms, then also $f(t_1, \ldots, t_n)$ is a term.*
  - *Nothing else is a term.*

- *An* τ-atomic formula *is any τ-word of the form $R(t_1, \ldots, t_n)$, where $t_1, \ldots, t_n$ are terms and R is a predicate symbol of arity n.*

- *A* τ-well-formed formula *(wff for short) is a τ-word built inductively as follows:*

  - *every atomic formula is a wff.*

– *If ϕ, ψ are wff and x is a variable, then expressions of the form*

$$\neg\phi \quad \phi \wedge \psi \quad \phi \vee \psi \quad \phi \to \psi \quad \forall x.\phi$$

are wff.

– *Nothing else is a wff.*

In the following, if clear from the context, we omit the suffix "τ-" in "τ-terms", "τ-expressions" etc.

Let $x \in \mathrm{Var}$ and ϕ be a wff. We say that *x is free in ϕ* if:

- $\phi = R(t_1, \dots, t_n)$ is an atomic formula and x occurs in some term t_i.
- ϕ has the form $\neg\psi$ and x is free in ψ.
- ϕ has one the following forms: $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$, $\psi_1 \to \psi_2$, and x is free in ψ_1 or in ψ_2.
- ϕ has the form $\forall z.\psi$ and $x \neq z$ and x is free in ψ.

We say that *x is bound in ϕ* iff it is not free in ϕ.

A *sentence* is a formula without free variables.

6.1.1 Structures

Definition 6.1.3. *A structure $\mathcal{M}$ of type τ is a tuple $(M, R^{\mathcal{M}}, f^{\mathcal{M}}, c^{\mathcal{M}})_{R,f,c\in\tau}$ satisfying the following conditions:*

- *M is a non-empty set, called the universe of $\mathcal{M}$.*
- *$R^{\mathcal{M}}$ is a subset of M^n, for any n-ary predicate symbol $R \in \tau$.*
- *$f^{\mathcal{M}}$ is a function from M^n to M, for any n-ary function symbol $f \in \tau$.*
- *$c^{\mathcal{M}}$ is an element of M, for any constant symbol $c \in \tau$.*

We denote by Str_τ the class of all structures of type τ and by Str^*_τ the class of all structures of type τ whose universe has cardinality at least 2.

A function $\rho : \mathrm{Var} \to M$ from the set Var of variables to the universe M of a structure $\mathcal{M}$ is called *environment*.

Let ϕ be a wff, $\mathcal{M}$ be a structure and ρ be an environment. Roughly speaking, we say that *$\mathcal{M}$ satisfies ϕ with ρ* (notation $\mathcal{M} \models_\rho \phi$) when the translation of ϕ determined by $\mathcal{M}$, where all variables x_i are translated as $\rho(x_i)$ whenever they occur free in ϕ, is true. The formal definition of satisfaction is not difficult but it is rather cumbersome ([30] pp. 83-84).

Let ϕ be a sentence and $\mathcal{M}$ be a structure. It can be proved that if $\mathcal{M} \models_{\rho'} \phi$ for a particular environment ρ', then $\mathcal{M} \models_\rho \phi$ for any environment ρ. So, if ϕ is a sentence, we can drop the subscript ρ in $\mathcal{M} \models_\rho \phi$.

Definition 6.1.4. *Let ϕ be a sentence and Σ be a set of sentences. We say that Σ logically implies ϕ, and we write $\Sigma \models \phi$, if and only if for every structure $\mathcal{M}$, if $\mathcal{M} \models \psi$ for all $\psi \in \Sigma$, then $\mathcal{M} \models \phi$.*

A sentence ϕ is called *valid* if $\varnothing \models \phi$.

Definition 6.1.5. *Let Σ be a set of sentences. We say that $\mathcal{M}$ is a model of Σ iff for every $\psi \in \Sigma$, $\mathcal{M} \models \psi$.*

Let ϕ be a wff whose free variables are $x_1, \dots x_n$. We have that, for any structure $\mathcal{M}$,

$$\text{for any environment } \rho, \mathcal{M} \models_\rho \phi \text{ if and only if } \mathcal{M} \models \forall x_1 \dots \forall x_n.\phi.$$

The sentence $\forall x_1 \dots \forall x_n.\phi$ is called the *universal closure of ϕ*.

A set of sentences Σ is a *theory* if it is closed under logical implication, i.e. if $\Sigma \models \phi$ implies $\phi \in \Sigma$. A theory Σ is called *complete* if, for any formula ϕ, $\phi \in \Sigma$ or $\neg\phi \in \Sigma$. The set of all true sentences of a structure is always a complete theory.

A theory is *consistent* if it is a proper subset of the set of sentences, *inconsistent* otherwise.

6.1.2 Homomorphisms of structures

Definition 6.1.6. *Let $\mathcal{M}, \mathcal{N}$ be structures of type τ. A function $h : M \to N$ is an homomorphism of structures if:*

- *for any n-ary predicate symbol R and $a_1, \dots, a_n \in M$,*

$$R^{\mathcal{M}}(a_1, \dots, a_n) \Rightarrow R^{\mathcal{N}}(h(a_1), \dots, h(a_n)).$$

- *For any n-ary function symbol f and $a_1, \dots, a_n \in M$,*

$$h(f^{\mathcal{M}}(a_1, \dots, a_n) = f^{\mathcal{N}}(h(a_1), \dots, h(a_n)).$$

- *For any constant symbol c,*

$$h(c^{\mathcal{M}}) = c^{\mathcal{N}}.$$

A *strong homomorphism of structures* is an homomorphism such that, for any n-ary relation R,

$$R^{\mathcal{M}}(a_1, \dots, a_n) \Leftrightarrow R^{\mathcal{N}}(h(a_1), \dots, h(a_n)).$$

A bijective strong homomorphism between structures $\mathcal{M}$ and $\mathcal{N}$ is called *isomorphism.* If such a function exists, we say that $\mathcal{M}$ and $\mathcal{N}$ are *isomorphic* (notation $\mathcal{M} \cong \mathcal{N}$).

6.1.3 Products of structures

Definition 6.1.7. *Let X be a non-empty set. A set $\mathcal{F} \subseteq \mathcal{P}(X)$ is a* filter *on X if the following conditions hold:*

- $X \in \mathcal{F}$.
- $x, y \in \mathcal{F}$ *implies* $x \cap y \in \mathcal{F}$.
- $x \in \mathcal{F}$ *and* $x \subseteq y$ *imply* $y \in \mathcal{F}$.

A proper filter not containing 0 *which is not properly contained in any other filter is called* ultrafilter .

Given elements a, b of an arbitrary product $\prod_{i \in I} A_i$, the set $[a = b] = \{i \in I : a_i = b_i\}$ is called the *equalizer* of a and b.
In the following, $(\mathcal{M}_i)_{i \in I}$ is a non-empty set of structures of type τ. $a(i)$ denotes the i^{th} coordinate of an element $a \in \prod_i M_i$.

Definition 6.1.8. *Let $\mathcal{F}$ be a filter on I. We define an equivalence relation $\theta_{\mathcal{F}}$ on $\prod_i M_i$ as follows:*

$$(a, b) \in \theta_{\mathcal{F}} \text{ iff } [\![a = b]\!] \in \mathcal{F}.$$

The set of equivalence classes of $\prod_i M_i$ with respect to $\theta_{\mathcal{F}}$ is denoted by $\prod_i M_i/\mathcal{F}$.

If $(a_1, b_1), \ldots, (a_n, b_n)$ are all in $\theta_{\mathcal{F}}$, then $(f(a_1, \ldots, a_n), f(b_1, \ldots, b_n)) \in \theta_{\mathcal{F}}$ for any n-ary function symbol f of τ.

Definition 6.1.9. *Let $(\mathcal{M}_i)_{i \in I}$ be structures and $\mathcal{F}$ be a filter on I. The* reduced product *of $(\mathcal{M}_i)_{i \in I}$ w.r.t. $\mathcal{F}$ is the structure $\mathcal{N} = \prod_{i \in I} \mathcal{M}_i/\mathcal{F}$ defined as follows:*

$$\mathcal{N} = (\prod_{i \in I} M_i/\mathcal{F}, R^{\mathcal{N}}, f^{\mathcal{N}}, c^{\mathcal{N}})$$

$$\begin{array}{rcl} (a_1, \ldots, a_n) \in R^{\mathcal{N}} & \text{iff} & \{i \in I : (a_1(i), \ldots, a_n(i)) \in R^{\mathcal{M}_i}\} \in \mathcal{F} \\ f^{\mathcal{N}}(a_1/\mathcal{F}, \ldots, a_n/\mathcal{F}) & = & f^{\prod \mathcal{M}_i}(a_1, \ldots, a_n)/\mathcal{F} \\ c^{\mathcal{N}} & = & c^{\prod \mathcal{M}_i}/\mathcal{F} \end{array}$$

If $\mathcal{F}$ is the trivial filter $\{I\}$, then $\prod_i \mathcal{M}_i/\mathcal{F}$ is denoted by $\prod_i \mathcal{M}_i$ and is called *direct product of* $(\mathcal{M}_i)_{i \in I}$. If $\mathcal{U}$ is an ultrafilter, then $\prod_i \mathcal{M}_i/\mathcal{U}$ is called *ultraproduct.*

6.2 Universal Algebra

In this section we follow [23].

Definition 6.2.1. *Let τ be an algebraic type (that is, a first-order type without relation symbols). An algebra $\mathbf{A}$ of type τ is a structure of type τ, i.e., $\mathbf{A} = (A, f^{\mathbf{A}}, c^{\mathbf{A}})_{f,c\in\tau}$ where A is a non-empty set, $f^{\mathbf{A}}$ is a function from A^n to A and $c^{\mathbf{A}}$ is a fixed element of $\mathbf{A}$ (for every n-ary function symbol f and constant c).*

An algebra whose underlying set is a singleton is called *trivial algebra*.

According to Definition 6.2.1, all of the following well-known structures are algebras: semigroups, monoids, groups, semirings, rings, modules, lattices, boolean algebras.

Definition 6.2.2. *An algebra $\mathbf{B}$ of type τ is a* subalgebra *of an algebra $\mathbf{A}$ (notation $\mathbf{A} \leq \mathbf{B}$) if $B \subseteq A$, $f^{\mathbf{B}}(b_1, \ldots, b_n) = f^{\mathbf{A}}(b_1, \ldots, b_n)$, for any $f \in \tau$ and elements $b_1, \ldots, b_n \in B$, and $c^{\mathbf{B}} = c^{\mathbf{A}}$ for any constant symbol $c \in \tau$.*

Definition 6.2.3. *Let $\mathbf{A}, \mathbf{B}$ be algebras on the same type τ. A function $h : A \to B$ is a* homomorphism of algebras *if*

$$h(f^{\mathbf{A}}(a_1, \ldots, a_n)) = f^{\mathbf{B}}(h(a_1), \ldots, h(a_n))$$

for all $f \in \tau$ and $a_1, \ldots, a_n \in A$.

An injective homomorphism is called *embedding*. An injective and surjective homomorphism is an *isomorphism*.

Definition 6.2.4. *Let $\mathbf{A}$ be an algebra of type τ. An equivalence relation θ on $\mathbf{A}$ is a* congruence *if, for any n-ary symbol f and elements $a_1, \ldots, a_n, b_1 \ldots, b_n$ of A,*

$$(a_1, b_1), \ldots, (a_n, b_n) \in \theta \;\Rightarrow\; (f^{\mathbf{A}}(a_1, \ldots, a_n), f^{\mathbf{A}}(b_1, \ldots, b_n)) \in \theta$$

Notation: if clear from the context, we denote $(a, b) \in \theta$ by $a\theta b$.

Given a, b elements of A, we denote by $\theta(a, b)$ the least congruence on $\mathbf{A}$ such that a and b are in the same equivalence class; a congruence of the form $\theta(a, b)$ is called a *principal congruence*. Given an arbitrary set $X \subseteq A$, we denote by $\theta(X)$ the least congruence such that all elements of X are in the same equivalence class.

We denote by Δ and ∇ respectively the congruences $\{(x, y) \in A \times A : x = y\}$ and $\{(x, y) \in A \times A : x, y \in A\}$. An algebra whose congruences are only Δ and ∇ is called *simple*.

Given congruences θ, θ' on $\mathbf{A}$, the least congruence containing θ and θ' is denoted by $\theta \vee \theta'$. The greatest congruence contained in θ and in θ' is exactly $\theta \cap \theta'$.

Definition 6.2.5. *Let θ be a congruence on $\mathbf{A}$. The quotient algebra of $\mathbf{A}$ by θ is the algebra $\mathbf{A}/\theta$ whose universe is A/θ and whose operations are defined as follows: for any n-ary symbol f and elements $a_1, \ldots, a_n$ of A,*

$$f^{\mathbf{A}/\theta}(a_1/\theta, \ldots, a_n/\theta) = f^{\mathbf{A}}(a_1 \ldots, a_n)/\theta.$$

Given an homomorphism $h:\ \mathbf{A} \to \mathbf{B}$, the *kernel* of h, ($ker(h)$ for short), is the set

$$ker(h) = \{(a,b) \in A \times A:\ h(a) = h(b)\}.$$

For every homomorphism $h : \mathbf{A} \to \mathbf{B}$, $ker(h)$ is a congruence on $\mathbf{A}$. Conversely, any congruence on $\mathbf{A}$ is the kernel of a homomorphism from $\mathbf{A}$ to an appropriate algebra $\mathbf{B}$.

Definition 6.2.6. *Given algebras* $\mathbf{A}, \mathbf{B}$ *of the same type* τ*, the* direct product $\mathbf{A} \times \mathbf{B}$ *of* $\mathbf{A}$ *and* $\mathbf{B}$ *is the algebra whose universe is the set* $A \times B$ *and whose operations are defined as follows: for all* $a_1, \ldots, a_n \in A,\ b_1, \ldots, b_n \in B$ *and* n*-ary symbol* $f \in \tau$*,*

$$f^{\mathbf{A}\times\mathbf{B}}((a_1,b_1), \ldots, (a_n,b_n)) = (f^{\mathbf{A}}(a_1, \ldots, a_n), f^{\mathbf{B}}(b_1, \ldots, b_n)),$$

and, for any constant symbol $c \in \tau$*,*

$$c^{\mathbf{A}\times\mathbf{B}} = (c^{\mathbf{A}}, c^{\mathbf{B}}).$$

The map $\pi_A : A \times B \to A$, defined as $\pi_A(a,b) = a$ is an homomorphism from $\mathbf{A}\times\mathbf{B}$ to $\mathbf{A}$ and is called *projection on the first coordinate.* The map $\pi_B : A \times B \to B$, defined as $\pi_B(a,b) = b$ is an homomorphism from $\mathbf{A}\times\mathbf{B}$ to $\mathbf{B}$ and is called *projection on the second coordinate.*

All these definitions generalize to products of algebras on any set of indices $I \neq \varnothing$.

Definition 6.2.7. *A congruence* θ *on an algebra* $\mathbf{A}$ *is a* factor congruence *if there exists another congruence* $\bar{\theta}$ *such that* $\theta \cap \bar{\theta} = \Delta$ *and* $\theta \circ \bar{\theta} = \nabla$*, where* $\circ$ *is the composition of relations. In this case we say that* $(\theta, \bar{\theta})$ *is a* pair of complementary factor congruences*.*

If $(\theta, \bar{\theta})$ is a pair of complementary factor congruences, then $\mathbf{A}$ is isomorphic to $\mathbf{A}/\theta \times \mathbf{A}/\bar{\theta}$. Any decomposition of an algebra $\mathbf{A}$ as a Cartesian product is, up to isomorphism, isomorphic to $\mathbf{A}/\theta \times \mathbf{A}/\bar{\theta}$ for some pair of complementary factor congruences.

An algebra which is not isomorphic to the product of two non-trivial algebras is called *directly indecomposable.* Directly indecomposable algebras are exactly those with one pair of factor congruences (Δ, ∇).

Definition 6.2.8. *Given a non-empty set of algebras* $(\mathbf{A_i})_{i\in I}$*, we say that* $\mathbf{C}$ *is a subdirect product of* $\prod_i \mathbf{A}_i$ *if* $\mathbf{C}$ *is a subalgebra of* $\prod_i \mathbf{A}_i$ *and* $\pi_{A_i}(C) = A_i$ *for all* $i \in I$*.*

An embedding $h : \mathbf{A} \to \prod_{i\in I} \mathbf{A}_i$ is *subdirect* if $h(\mathbf{A})$ is a subdirect product of $\prod_{i\in I} \mathbf{A}_i$.

Definition 6.2.9. *An algebra* $\mathbf{A}$ *is* subdirectly irreducible *if for any subdirect embedding* $h : \mathbf{A} \to \prod_{i\in I} \mathbf{A}_i$ *there is* $j \in I$ *such that* $\pi_j \circ h$ *is an isomorphism.*

The totally ordered lattice $\mathbf{3} = \{3, \wedge, \vee\}$ is an example of a subdirectly irreducible algebra: given $\mathbf{2} = \{2, \wedge, \vee\}$, $\mathbf{3}$ can be seen as a subalgebra of $\mathbf{2} \times \mathbf{2}$ such that $\pi(3) = 2$, but a cardinality reason shows that such π cannot be injective.

An algebra $\mathbf{A}$ is subdirectly irreducible iff it is trivial or it has a minimal congruence strictly containing Δ. This implies that any subdirectly irreducible algebra is directly indecomposable.

Theorem 6.2.10. *(Birkhoff, [23]) Every algebra is isomorphic to a subdirect product of subdirectly irreducible algebras.*

6.2.1 Varieties

Given a class $\mathcal{K}$ of algebras of type τ, we define:

$S(\mathcal{K})$, the class of the subalgebras of algebras in $\mathcal{K}$.
$H(\mathcal{K})$, the class of the homomorphic images of algebras in $\mathcal{K}$.
$P(\mathcal{K})$, the class of the direct products of algebras in $\mathcal{K}$.

Definition 6.2.11. *A nonempty class $\mathcal{K}$ of algebras of type τ is a* variety *if it closed under subalgebras, homomorphic images and direct products.*

Given a class of algebras $\mathcal{K}$, we denote by $\mathsf{V}(\mathcal{K})$ the variety generated by $\mathcal{K}$.

Theorem 6.2.12. *(Tarski) Given a class of algebras $\mathcal{K}$, $\mathsf{V}(\mathcal{K}) = HSP(\mathcal{K})$.*

Thanks to Theorem 6.2.10, the following fundamental result holds.

Theorem 6.2.13. *(Birkhoff) Every algebra in a variety V is isomorphic to a subdirect product of subdirectly irreducible members of V.*

A class of algebras K is called *equational* if there exists a set Σ of equations such that

$$K = \{\mathbf{A} : \mathbf{A} \text{ satisfies all equations of } \Sigma\}.$$

Theorem 6.2.14. *(Birkhoff, [23]) A class of algebras is a variety if and only if it is an equational class.*

6.2.2 Decomposition operators

Factor congruences can be characterized in terms of certain algebra homomorphisms called *decomposition operators* (see [51, Def. 4.32] for more details).

Definition 6.2.15. *Let $\mathbf{A}$ be an algebra of type τ. A* decomposition operator *on $\mathbf{A}$ is a function $f : A \times A \to A$ satisfying the following conditions:*

(D1) $f(x, x) = x$.

(D2) $f(f(x_{11}, x_{12}), f(x_{21}, x_{22})) = f(x_{11}, x_{22})$.

(D3) *f is an homomorphism from $\mathbf{A} \times \mathbf{A}$ into $\mathbf{A}$.*

There exists a bijective correspondence between a pair of complementary factor congruences and decomposition operations.

Proposition 6.2.16. *[51, Thm.4.33] Let $\mathbf{A}$ be an algebra of type τ. Given a decomposition operator $f : A^2 \to A$, the binary relations θ and $\bar{\theta}$, defined by:*

$$x \; \theta \; y \quad \textit{if, and only if,} \quad f(x,y) = y; \qquad x \; \bar{\theta} \; y \quad \textit{if, and only if,} \quad f(x,y) = x,$$

form a pair of complementary factor congruences. Conversely, given a pair $(\theta, \bar{\theta})$ of complementary factor congruences, the function f defined by:

$$f(x,y) = u \quad \textit{if, and only if,} \quad x\theta u\bar{\theta}y, \tag{6.1}$$

determines a decomposition operator on $\mathbf{A}$.

6.2.3 Boolean algebras

Definition 6.2.17. *A* boolean algebra $\mathbf{B}$ *is an algebra $(B, \wedge, \vee, ', 0, 1)$ where $\wedge, \vee$ are binary functions, $'$ is an unary function and $0, 1$ are constants satisfying the following identities, for all $x, y, z \in B$:*

- $x \wedge x = x, \quad x \vee x = x.$
- $x \wedge y = y \wedge x, \quad x \vee y = y \vee x.$
- $x \wedge (y \wedge z) = (x \wedge y) \wedge z, \quad x \vee (y \vee z) = (x \vee y) \vee z.$
- $x \wedge (x \vee y) = x, \quad x \vee (x \wedge y) = x.$
- $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z), \quad x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z).$
- $x \wedge 0 = 0, \quad x \vee 1 = 1.$
- $x \wedge x' = 0, \quad x \vee x' = 1.$

Given a set X, the structure $(\mathcal{P}(X), \cap, \cup, ', \varnothing, X)$ is a boolean algebra called *Boolean algebra of all subsets of X*. The boolean algebra of truth values is denoted by $\mathbf{2}$ and it is the simplest non-trivial boolean algebra.

Theorem 6.2.18 (Stone, [23] pag. 134)**.** $\mathbf{2}$ *is the only subdirectly irreducible boolean algebra.*

Theorem 6.2.19 (Stone,[23] pag. 134)**.** *Any boolean algebra $\mathbf{B}$ is a subdirect product of $\mathbf{2}^I$ for some set of indices I.*

6.3 Discriminator varieties

The notion of discriminator variety is one of the most important generalizations of Boolean algebras, since it is possible to use Boolean product representations (see [23], pp. 174-191).

Definition 6.3.1. *Let* $\mathbf{A}$ *be an algebra of type* τ *and* $f : A^n \to A$ *a function. We say that the term* $u(x_1, \dots, x_n)$ *of type* τ realizes f *in* $\mathbf{A}$ *if, for all* $a_1, \dots, a_n \in A$*,*

$$u^{\mathbf{A}}(a_1, \dots, a_n) = f(a_1, \dots, a_n).$$

Given a set A, the *switching function* $s : A^4 \to A$ is the function defined as follows:

$$s(a, b, x, y) = \begin{cases} x & \text{if } a = b. \\ y & \text{otherwise.} \end{cases}$$

Definition 6.3.2. *Let* $\mathcal{K}$ *be a class of algebras such that there is a term* u *that realizes the switching function for all* $\mathbf{A} \in \mathcal{K}$*. Then the variety* $\mathsf{V}(\mathcal{K})$ *is called* discriminator variety*.*

The following theorem equationally characterizes discriminator varieties.

Theorem 6.3.3. *(Vaggione [67]). A variety* V *of type* τ *is a discriminator variety iff there is a quaternary term* u *of type* τ *satisfying the following identities:*

(D1) $u(x, x, y, z) = y$.

(D2) $u(x, y, z, z) = z$.

(D3) $u(x, y, u(x, y, v, w), z) = u(x, y, v, z) = u(x, y, v, u(x, y, w, z))$.

(D4) $u(x, y, f(\vec{v}), f(\vec{w})) = f(u(x, y, v_1, w_1), \dots, u(x, y, v_n, w_n))$*, for any* n*-ary symbol* $f \in \tau$*.*

(D5) $u(x, y, x, y) = y$.

7

Factor algebras for first-order logic

The notion of discriminator variety is one of the keynote of Universal Algebra. It is the most successful generalization of Boolean algebras: it provides a common context for different kind of algebras (e.g., Boolean algebras, Post algebras, n-dimensional cylindric algebras). Discriminator varieties satisfy strong algebraic properties, such as the Boolean product representation. A lot of generalizations of the discriminator varieties has been proposed. *Factor varieties* are among them. They are the starting point of this chapter and were introduced in a restricted form in [58].

In Section 7.1 we recall the original definition of factor algebra and introduce a first generalization of this notion.

In Section 7.2 we define the factor algebras associated with a first-order type and show their relationship with the class of first-order structures. The class functions Str and Fa are defined in the same section. We also study the properties of the lattice of the congruences of a factor algebra by the notion of splitting pair. We characterize simple, subdirectly irreducible and directly indecomposable factor algebras through congruence properties and introduce rigid and pure factor algebras.

In Section 7.3 we define factor varieties and prove some results on the classes of their factor algebras.

In the final section, Section 7.4, we study in more details the class functions Str and Fa.

7.1 Preliminaries

We recall that in a discriminator variety V of type τ there is a τ-term u that realizes the discriminator function for any subdirectly irreducible $\mathbf{A} \in \mathsf{V}$ (see Section 6.3). Roughly speaking, the term u algebraize the equality relation. Factor varieties generalize discriminator varieties in the sense that the equality relation is substituted by an arbitrary binary relation R. More formally, a variety V is a factor variety if there is a quaternary term u, called factor term, satisfying the following condition for any subdirectly irreducible algebra $\mathbf{A} \in \mathsf{V}$: $u^{\mathbf{A}}(a, b, c, d) \in \{c, d\}$ for every $a, b, c, d \in A$. It is possible to show that V is a factor variety with a common factor term u iff, for any $\mathbf{A} \in \mathsf{V}$ and all $a, b \in A$, the binary function $u(a, b, -, -)$ is a decomposition operator (see [58]). Factor algebras had not been introduced only for a purpose of generalization. Factor algebras provide a general environment for studying concepts coming from fuzzy logic (Gödel algebras, product algebras) and quantum logic (Jauch-Piron orthomodular lattices with states).

7.1.1 A generalization of factor algebras

A factor variety is defined as the variety generated by a class of factor algebras with a common factor term u.

In the original definition (see [58]) an algebra $\mathbf{A}$ of type τ is called a *factor algebra* if there is a quaternary τ-term u satisfying the following condition, for every $a, b \in A^2$:

$$\forall xy.u^{\mathbf{A}}(a,b,x,y) = x \;\; \veebar \;\; \forall xy.u^{\mathbf{A}}(a,b,x,y) = y.$$

The factor term u defines the binary relation $R_u = \{(a,b) : u^{\mathbf{A}}(a,b,x,y) = x\}$ on A.

This definition can be naturally generalized to represent any finitary relation.

Definition 7.1.1. $\mathbf{A}$ *is a* ***factor algebra*** *if there is a $n+2$-ary τ-term u satisfying the following condition, for every $a_1, \ldots, a_n \in A^n$:*

$$\forall yz.u^{\mathbf{A}}(a_1,\ldots,a_n,y,z) = y \;\; \veebar \;\; \forall yz.u^{\mathbf{A}}(a_1,\ldots,a_n,y,z) = z.$$

Any factor term u defines a n-ary relation

$$R_u = \{(a_1,\ldots,a_n) : \; u^{\mathbf{A}}(a_1,\ldots,a_n,y,z) = y \text{ for all } y,z \in \mathbf{A}\}.$$

This relation is called the *factor relation of u*.

Definition 7.1.1 is just the first step of a broader generalization. In the forthcoming section we introduce another type of factor algebras, that can realize an arbitrary number of finitary relations. This naturally leads us to link them with first-order types.

7.2 Factor algebras of first-order types

Let τ be a generic type of first-order structures with some (possibly none) function symbols and with some (possibly none) relation symbols.

We denote by ν_τ the algebraic type defined as follows:

1. If g is an n-ary function symbol of type τ, then g is also an n-ary function symbol of type ν_τ;

2. If R is an n-ary relation symbol of type τ, then $\hat{R}$ is an $n+2$-ary function symbol of type ν_τ;

3. Nothing else belongs to ν_τ.

We denote by

$$\mathrm{Alg}(\nu_\tau) = \{g : \ g \text{ is a function symbol of type } \tau\}$$

and by

$$\mathrm{Rel}(\nu_\tau) = \{\hat{R} : \ R \text{ is a relation symbol of type } \tau\}.$$

To simplify the notation we say that the symbol $\hat{R}$ is a relation symbol.

Definition 7.2.1. *A **factor algebra of first-order type** τ, or simply a τ**-factor algebra**, is a ν_τ-algebra $\mathbf{A} = (A, g^{\mathbf{A}}, \hat{R}^{\mathbf{A}})_{g,R\in\tau}$ satisfying the following condition, for all $\bar{a} \in A^n$ and all relation symbols $R \in \tau$:*

$$\forall yz.\hat{R}(\bar{a}, y, z) = y \ \veebar \ \forall yz.\hat{R}(\bar{a}, y, z) = z.$$

FA_τ denotes the class of all τ-factor algebras, while FA_τ^* denotes the class of all τ-factor algebras of cardinality ≥ 2.

Given a τ-factor algebra $\mathbf{A}$, the *relational reduct of* $\mathbf{A}$ is the algebra $\mathrm{Rel}(\mathbf{A}) = (A, \hat{R}^{\mathbf{A}})_{R\in\tau}$ and the *algebraic reduct of* $\mathbf{A}$ is the algebra $\mathrm{Alg}(\mathbf{A}) = (A, g^{\mathbf{A}})_{g\in\tau}$.

I recall that Str_τ^* denotes the class of structures of type τ whose universe is a non-singleton set.

Example 7.2.2. If equality is in τ, then any element of FA_τ is a discriminator algebra. The term $\hat{R}$ that algebraizes the equality relation is called *discriminator term.*

Definition 7.2.3. *We define a class function* Str *from* Fa_τ^* *to* Str_τ^* *as follows:*

$$\mathbf{A} = (A, g^{\mathbf{A}}, \hat{R}^{\mathbf{A}}) \mapsto \mathrm{Str}(\mathbf{A}) = (A, g^{\mathrm{Str}(\mathbf{A})}, R^{\mathrm{Str}(\mathbf{A})})$$

where functions and relations of $\mathrm{Str}(\mathbf{A})$ *are defined as*

$$g^{\mathrm{Str}\mathbf{A}} = g^{\mathbf{A}}$$
$$(a_1, \ldots, a_n) \in R^{\mathrm{Str}(\mathbf{A})} \ \textit{if} \ \forall xy.\hat{R}^{\mathbf{A}}(a_1, \ldots, a_n, x, y) = x.$$

Vice versa, we define a class function Fa *from* Str_τ^* *to* Fa_τ^**:*

$$\mathcal{M} = (M, g^{\mathcal{M}}, R^{\mathcal{M}}) \mapsto \mathrm{Fa}(\mathcal{M}) = (M, g^{\mathrm{Fa}(\mathcal{M})}, \hat{R}^{\mathrm{Fa}(\mathcal{M})})$$

where functions of $\mathrm{Fa}(\mathbf{A})$ *are defined as*

$$g^{\mathrm{Fa}(\mathcal{M})} = g^{\mathcal{M}}$$
$$\hat{R}^{\mathrm{Fa}(\mathcal{M})}(a_1, \ldots, a_n, x, y) = \begin{cases} x & \textit{if } (a_1, \ldots, a_n) \in R^{\mathcal{M}}. \\ y & \textit{if } (a_1, \ldots, a_n) \notin R^{\mathcal{M}}. \end{cases}$$

The following straightforward proposition builds a bridge between logic and the theory of factor algebras.

Proposition 7.2.4. *Class functions* Str *and* Fa *define bijective correspondences between the classes* Str^*_τ *and* Fa^*_τ.

In particular, for any structure $\mathcal{M}$ *and factor algebra* $\mathbf{A}$, $\mathrm{Str}(\mathrm{Fa}(\mathcal{M})) = \mathcal{M}$ *and* $\mathrm{Fa}(\mathrm{Str}(\mathbf{A})) = \mathbf{A}$.

The correspondence fails for structures of cardinality 1: let $\mathbf{C}_1$ and $\mathbf{C}_2$ be structures whose universe is a singleton set $\{x\}$ and whose type is the mono unary type $\tau = \{R\}$. If $\mathbf{C}_1 \models R(x)$ but $\mathbf{C}_2 \not\models R(x)$, then we have two non-isomorphic structures, while there is only one trivial factor algebra. In general, in fact, there exists different τ-structures whose universe is a singleton set, while we have just one trivial factor algebra.

If not explicitly stated, in the following we will consider only non-singleton structures and non-trivial factor algebras.

7.2.1 Congruences of factor algebras

In this section we develop the basics of the theory of congruences of factor algebras. The keynote is the notion of unsplitting pair: roughly speaking, an ordered pair (b, c) of a factor algebra $\mathbf{A}$ is unsplitting if we cannot "distinguish" b and c in the structure $\mathrm{Str}(\mathbf{A})$ by using relational symbols. With this notion we can compare the congruences of a factor algebra $\mathbf{A}$ to those of its algebraic reduct $\mathrm{Alg}(\mathbf{A})$ and give characterization of simple, subdirectly irreducible and directly indecomposable factor algebras.

Notation 2. *: if* $\bar{a} = a_1, \dots, a_n$ *is a sequence, then* $\bar{a}[b/i]$ *denotes the sequence*

$$a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n.$$

In the following $\mathbf{A}$ is factor algebra of type τ.

Definition 7.2.5. *We say that an ordered pair* $(b, c) \in A^2$ splits $\mathbf{A}$ *if there exists a relation symbol* $R \in \tau_n$ *and* $\bar{a} \in A^n$ *such that*

$$\hat{R}(\bar{a}[b/i], x, y) = \hat{R}(\bar{a}[c/i], y, x), \quad \textit{for all } x, y \in \mathbf{A}.$$

An ordered pair is unsplitting if it does not split $\mathbf{A}$. *We denote by* $\Upsilon_{\mathbf{A}}$ *the set of all unsplitting ordered pairs of* $\mathbf{A}$.

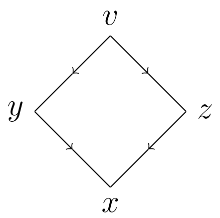
Equivalently to Definition 7.2.5, a pair (b, c) is unsplitting if for every n-ary relation R, for every $a_1, \dots, a_n, x, y \in A$ and for every index $1 \le i \le n$, we have that

$$\hat{R}(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n, x, y) = \hat{R}(a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_n, x, y). \quad (7.1)$$

Given a structure $\mathcal{M}$, a pair (b, c) is unsplitting if for every relation $R \in \tau_n$, for every $a_1, \dots, a_n \in M$ and for every index $1 \le i \le n$, we have that

$$(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n) \in R \text{ iff } (a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_n) \in R. \quad (7.2)$$

Example 7.2.6. Let $V = (\{a, b, c, d\}, <)$ be the strict partial order represented in the following diagram. The type is $\tau = \{<\}$: we stress that equality is not present.



Given any pair of elements $(a_1, a_2) \in V \times V$, we have $a_1 < y \Leftrightarrow a_1 < z$ and $y < a_2 \Leftrightarrow z < a_2$. So (y, z) is an unsplitting pair of V. The pair (x, y) is an example of splitting pair ($x < y$ holds but $y < y$ does not).

Lemma 7.2.7. *The set* $\Upsilon_{\mathbf{A}}$ *of all unsplitting pairs is a congruence on the algebra* $\mathrm{Rel}(\mathbf{A})$.

Proof. By Equation 7.1 it easily follows that $\Upsilon_{\mathbf{A}}$ is an equivalence relation.

Let $b_1 \Upsilon_{\mathbf{A}} c_1, \dots, b_n \Upsilon_{\mathbf{A}} c_n, x_1 \Upsilon_{\mathbf{A}} y_1, x_2 \Upsilon_{\mathbf{A}} y_2$ and $\hat{R}$ be a n-ary relation. We have that, thanks to Equation 7.2,

$$\begin{aligned} (b_1, b_2 \dots, b_n) &\in R^{\mathrm{Str}(\mathbf{A})} \Leftrightarrow \\ (c_1, b_2 \dots, b_n) &\in R^{\mathrm{Str}(\mathbf{A})} \Leftrightarrow \\ (c_1, c_2 \dots, b_n) &\in R^{\mathrm{Str}(\mathbf{A})} \Leftrightarrow \\ &\dots \\ (c_1, c_2 \dots, c_n) &\in R^{\mathrm{Str}(\mathbf{A})} \end{aligned}$$

It follows that $\hat{R}(b_1, \dots, b_n, x_1, x_2) \Upsilon_{\mathbf{A}} \hat{R}(c_1, \dots, c_n, y_1, y_2)$. □

Lemma 7.2.8. *Let* $R \in \tau_n$, *and* $\bar{a}, \bar{b} \in A^n$. *If* $\hat{R}(\bar{a}, x, y) = \hat{R}(\bar{b}, y, x)$ *with* $x \neq y$, *then there exists* $1 \leq k \leq n$ *such that* (a_k, b_k) *splits* $\mathbf{A}$.

Proof. Let i be the least index such that $a_i \neq b_i$. If $\hat{R}(\bar{a}, x, y) = \hat{R}(\bar{a}[b_i/i], y, x)$ then the pair (a_i, b_i) splits $\mathbf{A}$. Otherwise, we have $\hat{R}(\bar{a}, x, y) = \hat{R}(\bar{a}[b_i/i], x, y)$; thus $\hat{R}(\bar{a}[b_i/i], x, y) = \hat{R}(\bar{b}, y, x)$. We repeat the reasoning with the sequences $\bar{a}[b_i/i]$ and $\bar{b}$. We now find the least $j > i$ such that $a_j = a[b_i/i]_j \neq b_j$. Iterating the reasoning, if we do not find a pair splitting $\mathbf{A}$, we get the contradiction $\hat{R}(\bar{b}, x, y) = \hat{R}(\bar{b}, y, x)$. □

Let $S \subseteq A \times A$ be a relation. We say that S *splits* $\mathbf{A}$ if there exists $(c, d) \in S$ splitting $\mathbf{A}$ (i.e., $S \not\subseteq \Upsilon_{\mathbf{A}}$). If $B \subseteq A$ we say that B *splits* $\mathbf{A}$ if $B \times B$ splits $\mathbf{A}$.

The following lemma describes the principal congruences $\theta^{\mathbf{A}}(a, b)$ of $\mathbf{A}$ by considering its splitting property and the congruence $\theta^{\mathrm{Alg}(\mathbf{A})}$ of the algebraic reduct $\mathrm{Alg}(\mathbf{A})$.

Lemma 7.2.9. *Let* $\mathbf{A}$ *be a* τ*-factor algebra,* a *and* b *distinct elements of* $\mathbf{A}$ *and* $\theta^{\mathrm{Alg}(\mathbf{A})}(a,b)$ *the principal congruence on* $\mathrm{Alg}(\mathbf{A})$ *generated by* a, b*. Then the principal congruence* $\theta^{\mathbf{A}}(a,b)$ *on* $\mathbf{A}$ *generated by* a, b *satisfies the following condition:*

$$\theta^{\mathrm{Alg}(\mathbf{A})}(a,b) \subseteq \theta^{\mathbf{A}}(a,b)$$

In particular, if $\theta^{\mathrm{Alg}(\mathbf{A})}(a,b)$ *splits* $\mathbf{A}$*, then* $\theta^{\mathbf{A}}(a,b) = \nabla$*. Otherwise,* $\theta^{\mathrm{Alg}(\mathbf{A})}(a,b) = \theta^{\mathbf{A}}(a,b)$

Proof. The condition $\theta^{\mathrm{Alg}(\mathbf{A})}(a,b) \subseteq \theta^{\mathbf{A}}(a,b)$ is a straightforward consequence of the fact that the type of $\mathrm{Alg}(\mathbf{A})$ is contained into ν_τ.

Now let $\phi = \theta^{\mathrm{Alg}(\mathbf{A})}(a,b)$. If ϕ splits $\mathbf{A}$, then there exist $R \in \tau_n$, $\bar{a} \in A^n$ and $(c,d) \in \phi$ such that, for example, $x_0 = \hat{R}^{\mathbf{A}}(\bar{a}[c/i], x_0, x_1)\ \phi\ \hat{R}^{\mathbf{A}}(\bar{a}[d/i], x_0, x_1) = x_1$. Since x_0, x_1 can be arbitrarily chosen, this gives the conclusion.

If ϕ does not split $\mathbf{A}$, then by Lemma 7.2.8 $\bar{a}\phi\bar{b}$, $x_0\phi y_0$ and $x_1\phi y_1$ imply that $\hat{R}^{\mathbf{A}}(\bar{a}, x_0, x_1) = x_i\phi y_i = \hat{R}^{\mathbf{A}}(\bar{b}, y_0, y_1)$. ☐

Corollary 7.2.10. *Let* $\mathbf{A}$ *be a factor algebra. Every proper congruence of* $\mathbf{A}$ *is contained within* $\Upsilon_{\mathbf{A}}$ *(In symbols,* $\bigcup \mathrm{Con}^*(\mathbf{A}) \subseteq \Upsilon_{\mathbf{A}}$*). If* $\Upsilon_{\mathbf{A}}$ *is proper and* $\Upsilon_{\mathbf{A}} \in \mathrm{Con}(\mathbf{A})$*, then* $\Upsilon_{\mathbf{A}}$ *is the unique coatom of* $\mathrm{Con}(\mathbf{A})$*.*

Factor algebras in which any pair of elements is unsplitting are called rigid and have important properties: every directly decomposable factor $\mathbf{A}$ is such.

Definition 7.2.11. *A factor algebra* $\mathbf{A}$ *is rigid if* $\Upsilon_{\mathbf{A}} = A \times A$*.*

In other words, $\mathbf{A}$ is rigid iff for every relational symbol $\mathrm{R} \in \tau_n$, we have that $\mathrm{R}^{\mathrm{Str}(\mathbf{A})} = A^n$ or $\mathrm{R}^{\mathrm{Str}(\mathbf{A})} = \varnothing$. In such a case we say that $\mathrm{Str}(\mathbf{A})$ is rigid. Analogously, a structure $\mathcal{M}$ is rigid if $\mathrm{Fa}(\mathcal{M})$ is a rigid factor algebra.

Proposition 7.2.12. *If* $\mathbf{A}$ *is directly decomposable then* $\mathbf{A}$ *is rigid.*

Proof. Let $\mathbf{A}$ be directly decomposable. Then there exists a pair $(\phi, \bar{\phi})$ of non-trivial complementary factor congruences. By Lemma 7.2.9 and $\phi, \bar{\phi} \neq \nabla$, we have $\phi \cup \bar{\phi} \subseteq \Upsilon_{\mathbf{A}}$. Assume, by the way of contradiction, that $\mathbf{A}$ is not rigid. Then there exist $c, d \in A$ such that (c,d) splits $\mathbf{A}$. This means that $\hat{R}^{\mathbf{A}}(\bar{a}[c/i], x, y) = \hat{R}^{\mathbf{A}}(\bar{a}[d/i], y, x)$ for some $R \in \tau_n$ and $\bar{a} \in A^n$. Since $\phi, \bar{\phi} \subseteq \Upsilon_{\mathbf{A}}$ and $\phi \circ \bar{\phi} = \nabla$, then there exists a unique z such that $c\phi z\bar{\phi}d$. From $(c,z), (z,d) \in \Upsilon_{\mathbf{A}}$ we get the contradiction $(c,d) \in \Upsilon_{\mathbf{A}}$. ☐

The following proposition gives characterizations of simple, subdirectly irreducible and directly indecomposable factor algebras in terms of properties of their congruences.

Proposition 7.2.13. *Let* $\mathbf{A}$ *be a factor algebra of cardinality* > 2 *and* $\mathbf{B} = \mathrm{Alg}(\mathbf{A})$*. Then:*

(1) $\mathbf{A}$ *is simple iff every proper principal congruence on* $\mathbf{B}$ *splits* $\mathbf{A}$*.*

(2) $\mathbf{A}$ *is subdirectly irreducible and non-simple iff there exists a proper principal congruence* $\phi \in \mathrm{Con}(\mathbf{B})$*, that satisfies the following conditions:*

(a) ϕ *is an atom of* $\mathrm{Con}(\mathbf{B})$*;*

(b) ϕ *does not split* $\mathbf{A}$*;*

(c) Every $\psi \in \mathrm{Con}(\mathbf{B}) \setminus [\phi, \nabla]$ *splits* $\mathbf{A}$*.*

(3) $\mathbf{A}$ *is directly indecomposable iff at least one of the following conditions is satisfied:*

(d) $\mathbf{A}$ *is not rigid;*

(e) $\mathbf{A}$ *is rigid and* $\mathbf{B}$ *is directly indecomposable.*

Proof. (1) By Lemma 7.2.9.

(2) Let $\mathbf{A}$ be s.i., but not simple. Then $\mathrm{Con}(\mathbf{A})$ has a unique atom $\phi \neq \nabla$. By Lemma 7.2.9 ϕ does not split $\mathbf{A}$. If ϕ were not an atom of $\mathrm{Con}(\mathbf{B})$, then every congruence $\Delta \subset \psi \subset \phi$ would be also a congruence of $\mathrm{Con}(\mathbf{A})$, contradicting that ϕ is an atom of $\mathrm{Con}(\mathbf{A})$. Then, for every $\psi \in \mathrm{Con}(\mathbf{B})$, either $\phi \subseteq \psi$ or $\phi \cap \psi = \Delta$. In this last case ψ must split $\mathbf{A}$. The opposite direction is trivial.

(3) If $\mathbf{A}$ rigid, then, by Lemma 7.2.9, $\mathrm{Con}(\mathbf{A}) = \mathrm{Con}(\mathbf{B})$: so $\mathbf{A}$ is directly indecomposable iff $\mathbf{B}$ is so. If $\mathbf{A}$ is not rigid, then by Lemma 7.2.12 it is not directly indecomposable.

□

7.2.2 Pure factor algebras

Definition 7.2.14. *A factor algebra* $\mathbf{A}$ *or a structure* $\mathcal{M}$ *is pure if its type* τ *does not have function symbols.*

We obviously have that $\mathbf{A} = \mathrm{Rel}(\mathbf{A})$ and that every non-empty subset of $\mathbf{A}$ is a subalgebra.
The following two corollaries describe in details the congruences of pure factor algebras.

Corollary 7.2.15. *Let* $\mathbf{A}$ *be a pure* τ*-factor algebra and* a, b *be distinct elements of* A*. Then we have that the principal congruence* $\theta^{\mathbf{A}}(a, b)$ *on* $\mathbf{A}$ *generated by* a, b *satisfies the following condition:*

$$\theta^{\mathbf{A}}(a,b) = \begin{cases} \nabla^{\mathbf{A}} & \textit{if } (a,b) \textit{ splits } \mathbf{A}; \\ \Delta^{\mathbf{A}} \cup \{(a,b),(b,a)\} & \textit{otherwise.} \end{cases}$$

Proof. If (a, b) splits $\mathbf{A}$, then it follows from Lemma 7.2.9. Otherwise, we consider the characterization of unsplitting pairs given by Fact 7.1. □

Corollary 7.2.16. *Let $\mathbf{A}$ be a pure factor algebra.*

(1) If ϕ is an equivalence relation which does not split $\mathbf{A}$, then $\phi \in \mathrm{Con}^*(\mathbf{A})$.

(2) If $\mathbf{A}$ is rigid, any equivalence relation is a congruence.

(3) $\mathrm{Con}(\mathbf{A})$ is an atomic lattice, whose atoms are the principal congruences.

(4) $\Upsilon_{\mathbf{A}} \in \mathrm{Con}(\mathbf{A})$. If $\mathbf{A}$ is not rigid, then $\Upsilon_{\mathbf{A}}$ is the unique coatom.

Proof. (1), (2) and (3) are simple consequences of Corollary 7.2.15.

By Lemma 7.2.7, $\Upsilon_{\mathbf{A}}$ is a congruence on $\mathbf{A}$. If $\mathbf{A}$ is not rigid, then it is not equal to ∇, so by Corollary 7.2.10 it is the unique coatom.

□

Here we give an analogous result of Proposition 7.2.13 for pure factor algebras.

Corollary 7.2.17. *Let $\mathbf{A}$ be a pure τ-factor algebra of cardinality > 2. Then:*

(1) $\mathbf{A}$ is simple iff $\Upsilon_{\mathbf{A}} = \Delta$.

(2) $\mathbf{A}$ is subdirectly irreducible iff either $\Upsilon_{\mathbf{A}} = \Delta$ or $\Upsilon_{\mathbf{A}} = \Delta \cup \{(a, b), (b, a)\}$ for some a, b iff $|\mathrm{Con}(\mathbf{A})| = 3$.

(3) $\mathbf{A}$ is directly indecomposable and rigid iff $\mathbf{A}$ is finite of prime cardinality.

(4) If $\mathbf{A}$ is rigid then its congruence lattice is the lattice of its equivalence relations.

Proof. (1) Trivial by Proposition 7.2.13.

(2) Trivial by Proposition 7.2.13.

(3) If $\mathbf{A}$ rigid, then every equivalence relation is a congruence on $\mathbf{A}$. The lattice of equivalence relations on A does not admit non-trivial pairs of complementary equivalences iff A is finite of prime cardinality.

(4) By Corollary 7.2.16.

□

7.2.2.1 Decidability of pure and rigid structures

This short paragraph is devoted to pure and rigid structures: in particular we show the decidability of their theories.

Lemma 7.2.18. *Let $\mathcal{M}$ be a rigid structure. For any formula ϕ, $\mathcal{M} \models \forall x.\phi \Leftrightarrow \exists x.\phi$.*

Proof. Let R be a relation symbol of arity n. $\mathcal{M} \models \exists x.R(a_1, \dots, a_{i-1}, x, a_{i+1}, \dots, a_n)$ iff there is $b \in M$ such that $\mathcal{M} \models_\rho R(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n)$. But $\mathcal{M}$ is rigid, so the truth value of $R(\bar{x})$ is independent of the choice of the elements of M replacing $\bar{x}$. A simple induction on the complexity of ϕ proves the lemma. □

Proposition 7.2.19. *Let $\mathcal{M}$ be a pure and rigid τ-structure. Then the theory of $\mathcal{M}$ is decidable.*

Proof. To check the truth value of ϕ, we put ϕ in prenex normal form φ and then check the truth value of ψ, the matrix of φ. For doing this, we only need to check the truth value of the relations that appear in ψ and then to apply the propositional calculus. Thanks to Lemma 7.2.18, the result we get is equal to the truth value of ϕ.

□

7.3 Factor varieties

Definition 7.3.1. *Let $\mathcal{K}$ be a class of factor algebras of type τ.Then the variety $\mathsf{V}(\mathcal{K})$ generated by $\mathcal{K}$ is called a τ-**factor variety**.*

Definition 7.3.2. *Let τ be a first order type. FA_τ denotes the factor variety generated by the τ-factor algebras.*

By extending Prop 3.4 of [58], we have the following axiomatic characterization of factor varieties.

Theorem 7.3.3. *Let τ be a type. A variety $\mathcal{V}$ of type τ is a factor variety iff for every relation symbol $R \in \tau$ the following equations hold:*

F1 $\hat{R}(\bar{x}, z, z) = z$;

F2 $\hat{R}(\bar{x}, \hat{R}(\bar{x}, x_{11}, x_{12}), \hat{R}(\bar{x}, x_{21}, x_{22})) = \hat{R}(\bar{x}, x_{11}, x_{22})$.

F3 $\hat{R}(\bar{x}, h(\bar{y}), h(\bar{z})) = h(\hat{R}(\bar{x}, y_1, z_1), \hat{R}(\bar{x}, y_2, z_2), \dots, \hat{R}(\bar{x}, y_k, z_k))$, *for every $h \in \nu_\tau$ of arity k.*

Proof. Let $\mathbf{A}$ be a subdirectly irreducible algebra of $\mathcal{K}$ and $\bar{a}$ a n-tuple in $\mathbf{A}$.

"$\Rightarrow$"

(F1) By definition of factor variety, in $\mathbf{A}$ we have that $\hat{R}^{\mathbf{A}}(\bar{a}, y, z)$ is equal to the value of the $n+1$ or the $n+2$ coordinate, so $\hat{R}^{\mathbf{A}}(\bar{a}, z, z) = z$ holds.

(F2) If $\mathbf{A} \models \forall xy.\hat{R}^{\mathbf{A}}(\bar{a}, x, y) = x$, then $\hat{R}^{\mathbf{A}}(\bar{a}, \hat{R}^{\mathbf{A}}(\bar{a}, x_{11}, x_{12}), \hat{R}^{\mathbf{A}}(\bar{a}, x_{21}, x_{22})) = \hat{R}^{\mathbf{A}}(\bar{a}, x_{11}, x_{21}) = x_{11} = \hat{R}^{\mathbf{A}}(\bar{a}, x_{11}, x_{22})$.
If $\mathbf{A} \models \forall xy.\hat{R}^{\mathbf{A}}(\bar{a}, x, y) = y$, then $\hat{R}^{\mathbf{A}}(\bar{a}, \hat{R}^{\mathbf{A}}(\bar{a}, x_{11}, x_{12}), \hat{R}^{\mathbf{A}}(\bar{a}, x_{21}, x_{22})) = \hat{R}^{\mathbf{A}}(\bar{a}, x_{21}, x_{22}) = x_{22} = \hat{R}^{\mathbf{A}}(\bar{a}, x_{11}, x_{22})$.

(F3) Let h be in ν_τ of arity k.
If $\mathbf{A} \models \forall xy.\hat{R}^{\mathbf{A}}(\bar{a}, x, y) = x$, then $\hat{R}^{\mathbf{A}}(\bar{x}, h^{\mathbf{A}}(\bar{y}), h^{\mathbf{A}}(\bar{z})) = h^{\mathbf{A}}(\bar{y}) = h^{\mathbf{A}}(\hat{R}^{\mathbf{A}}(\bar{x}, y_1, z_1), \hat{R}^{\mathbf{A}}(\bar{x}, y_2, z_2), \ldots, \hat{R}^{\mathbf{A}}(\bar{x}, y_k, z_k))$.
If $\mathbf{A} \models \forall xy.\hat{R}^{\mathbf{A}}(\bar{a}, x, y) = y$, then $\hat{R}^{\mathbf{A}}(\bar{x}, h^{\mathbf{A}}(\bar{y}), h^{\mathbf{A}}(\bar{z})) = h^{\mathbf{A}}(\bar{z}) = h^{\mathbf{A}}(\hat{R}^{\mathbf{A}}(\bar{x}, y_1, z_1), \hat{R}^{\mathbf{A}}(\bar{x}, y_2, z_2), \ldots, \hat{R}^{\mathbf{A}}(\bar{x}, y_k, z_k))$.

"$\Leftarrow$" By conditions F1, F2 and F3 the function $f(y, z) = \hat{R}(\bar{a}, y, z)$ is a decomposition operator. $\mathbf{A}$ is directly indecomposable, then we have that $\{(y, z) : \hat{R}^{\mathbf{A}}(\bar{a}, y, z) = y\} = A^2$ or $\{(y, z) : \hat{R}^{\mathbf{A}}(\bar{x}, y, z) = z\} = A^2$, i.e., $\mathbf{A}$ is a factor algebra. □

Given a factor variety V, V_{fa} is the class of its factor varieties: in the following we show some of its properties.

Corollary 7.3.4. *Given a factor variety* V*, every directly indecomposable algebra* $\mathbf{A} \in \mathsf{V}$ *is a factor algebra.*

Proof. In any directly indecomposable algebra $\mathbf{A} \in \mathsf{V}$, every function $\hat{R}(\bar{a}, -, -)$ is a trivial decomposition operator. So exactly one among $\mathbf{A} \models \forall xy.\hat{R}(\bar{a}, x, y) = x$ and $\mathbf{A} \models \forall xy.\hat{R}(\bar{a}, y, x) = y$ holds. □

Proposition 7.3.5. *In a factor variety* V*, the class* V_{fa} *is closed under subalgebras, ultraproducts and homomorphic images.*

Proof. V_{fa} is closed under subalgebras and ultraproducts images because it is a universal class. Let $g : \mathbf{A} \to \mathbf{B}$ be a onto homomorphism and $\mathbf{A}$ be a factor algebra. We have that, for any relational symbol $R \in \tau_n$ and sequence $\bar{a}$ of elements of A,

$$\hat{R}^{\mathbf{B}}(g(\bar{a}), g(x), g(y)) = g(\hat{R}^{\mathbf{A}}(\bar{a}, x, y)) = \begin{cases} g(x) \text{ if } \hat{R}^{\mathbf{A}}(\bar{a}, x, y) = x. \\ g(y) \text{ if } \hat{R}^{\mathbf{A}}(\bar{a}, x, y) = y. \end{cases}$$

Thanks to the fact that g is surjective, the conclusion follows.

□

The class V_{fa} in general is not a variety, because it usually fails to be closed under direct product.

Let $\mathbf{A}$, $\mathbf{B}$ be factor algebras whose type τ include the equality relation, algebraized by $\hat{R}$: let (x, y), (x, y') be pairs such that $y \neq y'$. Given $v \neq z$, we have

$$\hat{R}^{\mathbf{A}\times\mathbf{B}}((x, y), (x, y'), (v, v), (z, z)) = (\hat{R}^{\mathbf{A}}(x, x, v, z), \hat{R}^{\mathbf{B}}(x, y', v, z)) = (v, z).$$

This shows that $\mathbf{A} \times \mathbf{B}$ is not a factor algebra of type τ. This example is the well-known fact that discriminator algebras are not closed under direct product.

7.4 Properties of class functions Str and Fa

In this section we study in more details the class functions Str and Fa introduced in Definition 7.2.3. In particular we show that Str can be extended to a functor from the category of τ-structures with strong homomorphisms as arrows to the category of τ-factor algebras with homomorphisms.

In the second part we enlarge the domain of Str to all algebras of type ν_τ and then we prove some results of equivalence between structures defined using Str and Fa.

Lemma 7.4.1. *Let* $\mathbf{A}, \mathbf{B}$ *be* τ*-factor algebras and* $\mathcal{M}, \mathcal{N}$ τ*-structures.*

(1) If a function $g : A \to B$ *is a strong homomorphism from* $\mathrm{Str}(\mathbf{A})$ *to* $\mathrm{Str}(\mathbf{B})$*, then it is a homomorphism from* $\mathbf{A}$ *to* $\mathbf{B}$*. The converse holds if* g *is not constant.*

(2) If a function $h : M \to N$ *is a strong homomorphism from* $\mathcal{M}$ *to* $\mathcal{N}$ *then it is also a homomorphism from* $\mathrm{Fa}(\mathcal{M})$ *to* $\mathrm{Fa}(\mathcal{N})$*. The converse holds if* h *is not constant.*

Proof. (1)

Let g be a strong homomorphism from $\mathrm{Str}(\mathbf{A})$ to $\mathrm{Str}(\mathbf{B})$ and b, c in A such that $g(b) \neq g(c)$ (this is the only non-trivial case):

$$g(\hat{R}^{\mathbf{A}}(\bar{a}, b, c)) = \begin{cases} g(b) \Leftrightarrow \bar{a} \in R^{\mathrm{Str}(\mathbf{A})} \Leftrightarrow g(\bar{a}) \in R^{\mathrm{Str}(\mathbf{B})} \Leftrightarrow \hat{R}^{\mathbf{B}}(g(\bar{a}), g(b), g(c)) = g(b) \\ g(c) \Leftrightarrow \bar{a} \notin R^{\mathrm{Str}(\mathbf{A})} \Leftrightarrow g(\bar{a}) \notin R^{\mathrm{Str}(\mathbf{B})} \Leftrightarrow \hat{R}^{\mathbf{B}}(g(\bar{a}), g(b), g(c)) = g(b) \end{cases}$$

Let g be a non-constant homomorphism from $\mathbf{A}$ to $\mathbf{B}$.

$$\begin{array}{rcll} \bar{a} \in R^{\mathrm{Str}(\mathbf{A})} & \Leftrightarrow & \forall xy.\hat{R}^{\mathbf{A}}(\bar{a}, x, y) = x & \\ & \Leftrightarrow & \forall xy.\hat{R}^{\mathbf{B}}(\bar{g}(a), g(x), g(y)) = g(x) & \text{since } g \text{ is a homomorphism.} \\ & \Leftrightarrow & \forall zv.\hat{R}^{\mathbf{B}}(\bar{g}(a), z, v) = z & \text{since } \mathbf{B} \text{ is a non-trivial factor} \\ & & & \text{algebra and } g \text{ is not constant.} \\ & \Leftrightarrow & g(\bar{a}) \in R^{\mathrm{Str}(\mathbf{B})} & \end{array}$$

(2) Similar to (1).

□

Corollary 7.4.2. Fa *is a functor from the category* $\mathbb{ST}^*_\tau$ *of* τ*-structures (with strong homomorphisms as arrows) to the category* $\mathbb{FA}^*_\tau$ *of* τ*-factor algebras (with homomorphisms).*

7.4.1 An extension of Str

Let $\mathbf{A} = (A, g^{\mathbf{A}}, \hat{R}^{\mathbf{A}})_{g,R\in\tau}$ be a ν_τ algebra. We define the structure

$$\mathrm{Str}(\mathbf{A}) = (A, g^{\mathrm{Str}(\mathbf{A})}, R^{\mathrm{Str}(\mathbf{A})})_{g,R\in\tau}$$

where $g^{\mathrm{Str}(\mathbf{A})} = g^{\mathbf{A}}$ and $(a_1, \dots, a_n) \in R^{\mathrm{Str}(\mathbf{A})}$ iff $\forall xy.\hat{R}^{\mathbf{A}}(a_1, \dots, a_n, x, y) = x$.

Remark 7.4.3. In general it is not true that for a non-trivial ν_τ-algebra $\mathbf{A}$ we have $\mathrm{Fa}(\mathrm{Str}(\mathbf{A})) = \mathbf{A}$. Consider the non-trivial algebra $\mathbf{A}$ where $\hat{R}(a, x, y) = x$ and $\hat{R}(a, m, n) = n$. In $\mathrm{Str}(\mathbf{A})$ we have $a \notin R$, so in $\mathrm{Fa}(\mathrm{Str}(\mathbf{A}))$ it must be $\hat{R}(a, x, y) = y$.

Lemma 7.4.4. *Let $\mathcal{M}_i$, $i \in I$, be τ-structures and $\mathcal{F}$ a filter on I. Then*

$$\prod_{i\in I} \mathcal{M}_i/\mathcal{F} = \mathrm{Str}(\prod_{i\in I} \mathrm{Fa}(\mathcal{M}_i)/\mathcal{F})$$

Proof. Let a be an element of $\prod \mathcal{M}_i$: $\mathbf{a}$ denotes $a/\mathcal{F}$ and $a(i)$ denotes the i^{th} component of a.

The classical definition of reduced product says that

$$\prod \mathcal{M}_i/\mathcal{F} \models R(\mathbf{a_1}, \dots, \mathbf{a_n}) \text{ iff } \{i \in I : \mathcal{M}_i \models (a_1(i), \dots, a_n(i))\} \in \mathcal{F}.$$

Then we have:

$$\begin{array}{ll}
\{i \in I : \mathcal{M}_i \models R(a_1(i), \dots, a_n(i))\} \in \mathcal{F} & \Leftrightarrow \\
\{i \in I : \forall uv \in \mathcal{M}_i.\mathrm{Fa}(\mathcal{M}_i) \models \hat{R}(a_1(i), \dots, a_n(i), u, v) = u\} \in \mathcal{F} & \Leftrightarrow \\
\prod \mathrm{Fa}(\mathcal{M}_i) \models \forall \mathbf{xy}.\hat{R}(\mathbf{a_1}, \dots, \mathbf{a_n}, \mathbf{x}, \mathbf{y}) = \mathbf{x} & \Leftrightarrow \\
\mathrm{Str}(\prod \mathrm{Fa}(\mathcal{M}_i)) \models R(\mathbf{a_1}, \dots, \mathbf{a_n}) &
\end{array}$$

□

Lemma 7.4.5. *Given $\mathbf{A}_i$, $i \in I$, algebras of type ν_τ, and a filter $\mathcal{F}$ on I, we have that*

$$\mathrm{Str}(\prod_{i\in I} \mathbf{A}_i/\mathcal{F}) = \prod_{i\in I} \mathrm{Str}(\mathbf{A}_i)/\mathcal{F}$$

Proof. Let a be an element of $\prod \mathbf{A}_i$: $\mathbf{a}$ denotes $a/\mathcal{F}$ and $a(i)$ denotes the i^{th} component of a.

$$\begin{array}{ll}
\mathrm{Str}(\prod \mathbf{A}_i/\mathcal{F}) \models R(\mathbf{a_1}, \dots, \mathbf{a_n}) & \Leftrightarrow \\
\prod \mathbf{A}_i/\mathcal{F} \models \forall \mathbf{xy}.\hat{R}(\mathbf{a_1}, \dots, \mathbf{a_n}, \mathbf{x}, \mathbf{y}) = \mathbf{x} & \Leftrightarrow \\
\{i \in I : \forall uv \in \mathbf{A}_i.\mathbf{A}_i \models \hat{R}(a_1(i), \dots, a_n(i), u, v) = u\} \in \mathcal{F} & \Leftrightarrow \\
\{i \in I : \mathrm{Str}(\mathbf{A}_i) \models R(a_1(i), \dots, a_n(i))\} \in \mathcal{F} & \Leftrightarrow \\
\prod \mathrm{Str}(\mathbf{A}_i)/\mathcal{F} \models R(\mathbf{a_1}, \dots, \mathbf{a_n}) &
\end{array}$$

□

In general, it is not true that $\prod \mathrm{Fa}(\mathcal{M}_i) = \mathrm{Fa}(\prod \mathcal{M}_i)$.
Let $\mathcal{M}_1$ and $\mathcal{M}_2 = \{2\}$ be structures on the mono unary type $\tau = \{R\}$, where R is interpreted as always true in $\mathcal{M}_1$ and always false in $\mathcal{M}_2$. $\mathrm{Fa}(\mathcal{M}_1 \times \mathcal{M}_2)$ is the τ-factor algebra whose universe is 4 and its operation satisfies the identity $\hat{R}^{\mathrm{Fa}(\mathcal{M}_1\times\mathcal{M}_2}((a_1, a_2), x, y) = y$. $\mathrm{Fa}(\mathcal{M}_1) \times \mathrm{Fa}(\mathcal{M}_2)$ instead is only a ν_τ-algebra, because in it we have $\hat{R}^{\mathrm{Fa}(\mathcal{M}_1)\times\mathrm{Fa}(\mathcal{M}_2)}((a_1, a_2), (x_1, x_2), (y_1, y_2)) = (x_1, y_2)$.

If we consider ultrafilters, instead, an analogous statement holds:

Lemma 7.4.6. *Let $\mathcal{M}_i$, $i \in I$ be τ-structures and $\mathcal{U}$ be an ultrafilter on I. We have that:*

$$\prod_{i\in I} \mathrm{Fa}(\mathcal{M}_i)/\mathcal{U} = \mathrm{Fa}(\prod_{i\in I} \mathcal{M}_i/\mathcal{U})$$

Proof. Let a be an element of $\prod \mathcal{M}_i$: $\mathbf{a}$ denotes $a/\mathcal{U}$ and $a(i)$ the i^{th} component of a.
Let $a_1, \dots, a_n, b, c$ be elements of $\prod_{i\in I} M_i$ such that $\mathbf{b} \neq \mathbf{c}$. We show that the identity is an isomorphism of algebras.

$$\begin{array}{ll}
\prod \mathrm{Fa}(\mathcal{M}_i)/\mathcal{U} \models \hat{R}(\mathbf{a_1}, \dots, \mathbf{a_n}, \mathbf{b}, \mathbf{c}) = \mathbf{b} & \Leftrightarrow \\
\{i \in I : \forall xy \in M_i.\mathrm{Fa}(\mathcal{M}_i) \models \hat{R}(a_1(i), \dots, a_n(i), x, y = x\} \in \mathcal{U} & \Leftrightarrow \\
\{i \in I : \mathcal{M}_i \models R(a_1(i), \dots, a_n(i))\} \in \mathcal{U} & \Leftrightarrow \\
\prod \mathcal{M}_i/\mathcal{U} \models R(\mathbf{a_1}, \dots, \mathbf{a_n}) & \Leftrightarrow \\
\mathrm{Fa}(\prod \mathcal{M}_i/\mathcal{U}) \models \hat{R}(\mathbf{a_1}, \dots, \mathbf{a_n}, \mathbf{b}, \mathbf{c}) = \mathbf{b} &
\end{array}$$

$$\begin{array}{ll}
\prod \mathrm{Fa}(\mathcal{M}_i)/\mathcal{U} \models \hat{R}(\mathbf{a_1}, \dots, \mathbf{a_n}, \mathbf{b}, \mathbf{c}) = \mathbf{c} & \Leftrightarrow \\
\{i \in I : \forall xy \in M_i.\mathrm{Fa}(\mathcal{M}_i) \models \hat{R}(a_1(i), \dots, a_n(i), x, y = y\} \in \mathcal{U} & \Leftrightarrow \\
\{i \in I : \mathcal{M}_i \not\models R(a_1(i), \dots, a_n(i))\} \in \mathcal{U} & \Leftrightarrow \\
\prod \mathcal{M}_i/\mathcal{U} \not\models R(\mathbf{a_1}, \dots, \mathbf{a_n}) & \Leftrightarrow \\
\mathrm{Fa}(\prod \mathcal{M}_i/\mathcal{U}) \models \hat{R}(\mathbf{a_1}, \dots, \mathbf{a_n}, \mathbf{b}, \mathbf{c}) = \mathbf{c} &
\end{array}$$

By Proposition 7.3.5 $\prod \mathrm{Fa}(\mathcal{M}_i)/\mathcal{U}$ is a factor algebra, so there is nothing else to prove. □

Corollary 7.4.7. *Let $\mathbf{A}_i$ be τ-factor algebras and $\mathcal{U}$ an ultrafilter on I. Then*

$$\prod_{i\in I} \mathbf{A}_i/\mathcal{U} = \mathrm{Fa}(\prod_{i\in I} \mathrm{Str}(\mathbf{A}_i)/\mathcal{U})$$

Proof. We already know that for any factor algebra $\mathbf{A} = \mathrm{Fa}(\mathrm{Str}(\mathbf{A}))$. So we have that $\prod \mathbf{A}_i = \prod \mathrm{Fa}(\mathrm{Str}(\mathbf{A}_i))$ and $\prod \mathbf{A}_i/\mathcal{U} = \prod \mathrm{Fa}(\mathrm{Str}(\mathbf{A}_i))/\mathcal{U}$. Then by Lemma 7.4.6 $\prod \mathrm{Fa}(\mathrm{Str}(\mathbf{A}_i))/\mathcal{U} = \mathrm{Fa}(\prod \mathrm{Str}(\mathbf{A}_i)/\mathcal{U})$. □

8

Algebraic calculus for propositional logic

The starting point of this chapter is the bijective correspondence between structures and factor algebras stated in Fact 7.2.4.

In Section 8.1, we extend this correspondence to formulas and equations between terms, and we show that it has the following semantical meaning: given a non-singleton structure $\mathcal{M}$, its corresponding factor algebra $\mathbf{A}$ and a formula ϕ, $\mathcal{M} \models \phi$ iff $\mathbf{A} \models \phi^* = y_{\mathrm{t}}$, where $\phi^* = y_{\mathrm{t}}$ is the equation that corresponds to ϕ. This result allows us to use algebraic methods for studying first-order logic.

In Section 8.2, we consider the type τ_{CL} of classical propositional logic with a countable set of propositional variables. If $\mathbf{A}$ is a factor algebra of type τ_{CL}, then a propositional variable C corresponds in its algebraic translation to a decomposition operator $\hat{C} : A \times A \to A$ (without parameters). We define the variety $\mathsf{Fa}_{\tau_{CL}}$ generated by the τ_{CL}-factor algebras. We show that a formula ϕ is a tautology iff $\mathsf{Fa}_{\tau_{CL}}$ satisfies the equation $\phi^* = y_{\mathrm{t}}$.

We provide axioms for $\mathsf{Fa}_{\tau_{CL}}$, that turn out to be equational rules for an algebraic calculus for propositional logic. We also introduce a term rewriting system for this calculus, and we show that it is terminating and confluent.

In the last Section 8.3, we give a simplification of a result of Burris ([22]) and an application of factor algebras to graph theory.

8.1 A bridge between open formulas and equations

8.1.1 Good terms

In this technical paragraph we show that every term of type ν_τ is equivalent to a term of a particular form called *good* term. We do so in order to build a simple translation between equations of ν_τ terms and τ-formulas. After that this translation is defined in Section 8.1.2, the meaning of this paragraph will be clear (see Remark 8.1.5).

Definition 8.1.1. *Let τ be a first-order type and* Var *be a denumerable set of variables. A term of type ν_τ is called:*

(1) a base *term if it belongs to the set freely generated by* Var *using the function symbols of τ.*

(2) a good *term if it is generated by induction as follows: every base term is good; if t_1, t_2 are good terms, $R \in \tau_n$ is a relation symbol and $\bar{u}$ is a sequence of base terms of length n, then $\hat{R}(\bar{u}, t_1, t_2)$ is a good term.*

The important consequence of this definition is that in a good term of the form $\hat{R}(\bar{u}, t_1, t_2)$, any term of $\bar{u}$ is built by using only variables and functions symbols of τ.

Lemma 8.1.2. *Let V be a τ-factor variety, $R, g \in \tau_n$ and $t_1, \dots, t_n, u, v$ be good terms. Then there exist good terms w_1, w_2 such that $\mathsf{V} \models \hat{R}(t_1, \dots, t_n, u, v) = w_1$ and $\mathsf{V} \models g(t_1, \dots, t_n) = w_2$.*

Proof. The proof is by induction on the number of occurrences of relation symbols in the terms $t \equiv t_1, \dots, t_n$. If all these terms are base terms, then the conclusion is obvious. Let now $t_i \equiv \hat{S}(\bar{k}, h_1, h_2)$ $(S \in \tau)$ for some $1 \leq i \leq n$, where h_1, h_2 are good terms and $\bar{k}$ is a sequence of base terms. Then we can apply axioms F1-F3 of factor variety as follows:

$$\begin{array}{rcll}
\hat{R}(\bar{t}, u, v) & = & \hat{R}(t_1, \dots, \hat{S}(\bar{k}, h_1, h_2), \dots, u, v) & \\
& = & \hat{R}(\hat{S}(\bar{k}, t_1, t_1), \dots, \hat{S}(\bar{k}, h_1, h_2), \dots, \hat{S}(\bar{k}, u, u), \hat{S}(\bar{k}, v, v)) & \text{by F1} \\
& = & \hat{S}(\bar{k}, \hat{R}(t_1, \dots, h_1, \dots, u, v), \hat{R}(t_1, \dots, h_2, \dots, u, v)) & \text{by F3} \\
& = & \hat{S}(\bar{k}, v_1, v_2) &
\end{array}$$

By induction hypothesis, terms v_1, v_2 in $\hat{S}(\bar{k}, v_1, v_2)$ are both good terms that satisfy $\mathsf{V} \models \hat{R}(t_1, t_2, \dots, w_i, \dots, u, v) = v_i$. Such terms can be obtained because the sequence $t_1, \dots, w_i, \dots, t_n$ has a number of occurrences of relation symbols strictly less than $t_1, \dots, t_i, \dots, t_n$.

The case of a function symbol $g \in \tau$ is proved similarly. □

Proposition 8.1.3. *Let V be a τ-factor variety. For every term t there exists a good term t' such that $\mathsf{V} \models t = t'$.*

Proof. The proof is by induction on the pair (number of occurrences of relation symbols in the term t, complexity of the term t) with the lexicographic order.

If $t \equiv \hat{R}(\bar{w})$, then the number occurrences of relation symbols in each term w_i is strictly less than the number of occurrences in t. By induction hypothesis we have $\mathsf{V} \models w_i = v_i$ for some good term v_i. Then $\mathsf{V} \models \hat{R}(\bar{w}) = \hat{R}(\bar{v})$ and we can apply Lemma 8.1.2.

If $t \equiv g(\bar{w})$, then the complexity of w_i is less than the complexity of t. By induction hypothesis we have $\mathsf{V} \models w_i = v_i$ for some good term v_i. Then $\mathsf{V} \models g(\bar{w}) = g(\bar{v})$ and we can again apply Lemma 8.1.2 □

8.1.2 From equations to open formulas

In the next two sections we translate equations $t = u$ between ν_τ-terms and open formulas ϕ of type τ. We show that this translation has also a semantical meaning, i.e., a structure $\mathcal{M}$ and a factor algebra $\mathbf{A}$ corresponding in the sense of Fact 7.2.4, model respectively formulas and equations that correspond in the translation.

Definition 8.1.4. *Let $t = u$ be an identity: wlog we can assume t, u to be good terms thanks to Proposition 8.1.3. We define the propositional formula $\phi_{t,u}$ as follows:*

(1) If t, u are base terms then $\phi_{t,u} \equiv (\mathrm{t} = \mathrm{u})$.

(2) If $t \equiv \hat{R}(\bar{w}, t_1, t_2)$ and u is a base term, then

$$\phi_{t,u} = (R(\bar{w}) \to \phi_{t_1,u}) \wedge (\neg R(\bar{w}) \to \phi_{t_2,u}).$$

(3) If $t \equiv \hat{R}(\bar{w}, t_1, t_2)$ and $u = \hat{S}(\bar{v}, u_1, u_2)$, then we define

$$\phi_{t,u} = (S(\bar{v}) \to \phi_{t,u_1}) \wedge (\neg S(\bar{v}) \to \phi_{t,u_2}).$$

Remark 8.1.5. In this definition Proposition 8.1.3 is necessary. Without it, in points (2) and (3) it would not be clear the meaning of expressions $R(\bar{w})$ and $S(\bar{v})$. Instead, if we assume that t and u are good terms then we have that all terms of $\bar{w}$ and $\bar{v}$ are ordinary terms of type τ, so $R(\bar{w})$ and $S(\bar{v})$ are well-defined atomic formulas of type τ.

Now we can prove the first semantical result of equivalence between equations and open formulas.

Proposition 8.1.6. *Let $\mathbf{A}$ be a factor algebra, $\rho : \mathrm{Var} \to A$ be an environment and t, u be good terms. Then we have:*

$$\mathbf{A} \models_\rho t = u \;\Leftrightarrow\; \mathrm{Str}(\mathbf{A}) \models_\rho \phi_{t,u}.$$

Proof. (1) If t and u are base terms, then there is nothing to prove.

(2) $t = \hat{R}(\bar{w}, t_1, t_2)$ and u is a base term.

"$\Rightarrow$" Let $\mathbf{A} \models_\rho t = u$. Since $\mathbf{A}$ is a factor algebra, then either $\mathbf{A} \models_\rho t = t_1$ or $\mathbf{A} \models_\rho t = t_2$. Wlog, assume $\mathbf{A} \models_\rho t = t_1$. Then we have $\mathrm{Str}(\mathbf{A}) \models_\rho R(\bar{w})$, $\mathrm{Str}(\mathbf{A}) \not\models_\rho \neg R(\bar{w})$ and $\mathbf{A} \models_\rho t = u \;\Leftrightarrow\; \mathbf{A} \models_\rho t_1 = u$. By ind. hyp., the last condition is equivalent to $\mathrm{Str}(\mathbf{A}) \models_\rho \phi_{t_1,u}$. So $\phi_{t,u}$ holds in $\mathrm{Str}(\mathbf{A})$.

"$\Leftarrow$" Let $\mathrm{Str}(\mathbf{A}) \models_\rho \phi_{t,u}$. Then, wlog, $R(\bar{w})$: so $\phi_{t_1,u}$ holds, and by ind. hyp. also $\mathbf{A} \models_\rho t_1 = u$. Then we have $\mathbf{A} \models_\rho \hat{R}(\bar{w}, t_1, t_2) = t_1 = u$.

(3) Let $t = \hat{R}(\bar{w}, t_1, t_2)$ and $u = \hat{S}(\bar{v}, u_1, u_2)$.

"$\Rightarrow$" Let $\mathbf{A} \models_\rho t = u$ holds, so, wlog, $\mathbf{A} \models_\rho t = u_1$. Then we have $\mathrm{Str}(\mathbf{A}) \models_\rho S(\bar{v})$, $\mathrm{Str}(\mathbf{A}) \not\models_\rho \neg S(\bar{v})$ and $\mathbf{A} \models_\rho t = u \;\Leftrightarrow\; \mathbf{A} \models_\rho t = u_1$. By ind. hyp. the last condition is equivalent to $\mathrm{Str}(\mathbf{A}) \models_\rho \phi_{t,u_1}$. So $\phi_{t,u}$ holds in $\mathrm{Str}(\mathbf{A})$.

"$\Leftarrow$" Let $\mathrm{Str}(\mathbf{A}) \models_\rho \phi_{t,u}$. Then, wlog, $S(\bar{v})$: so $\phi_{\hat{R}(\bar{w},t_1,t_2),u_1}$ holds, and by ind. hyp. also $\mathbf{A} \models_\rho \hat{R}(\bar{w}, t_1, t_2) = u_1$. Then we have $\mathbf{A} \models_\rho \hat{R}(\bar{w}, t_1, t_2) = u_1 = \hat{S}(\bar{v}, u_1, u_2)$.

□

8.1.3 From open formulas to equations

Definition 8.1.7. *Let $y_{\mathrm{t}}, y_{\mathrm{f}} \notin \mathrm{Var}$ be two fresh variables and ϕ be a quantifier-free formula. We define the ν_τ-good term ϕ^* by induction as follows:*

$$\phi^* = \begin{cases} y_{\mathrm{t}} & \text{if } \phi \equiv \mathrm{true}; \\ y_{\mathrm{f}} & \text{if } \phi \equiv \mathrm{false}; \\ \hat{R}(\bar{t}, y_{\mathrm{t}}, y_{\mathrm{f}}) & \text{if } \phi \equiv R(\bar{t}) \text{ is an atomic formula;} \\ \psi^*[y_{\mathrm{t}}/y_{\mathrm{f}}; y_{\mathrm{f}}/y_{\mathrm{t}}] & \text{if } \phi \equiv \neg\psi; \\ \psi_2^*[\psi_1^*/y_{\mathrm{f}}] & \text{if } \phi \equiv \psi_1 \vee \psi_2; \\ \psi_2^*[\psi_1^*/y_{\mathrm{t}}] & \text{if } \phi \equiv \psi_1 \wedge \psi_2; \end{cases}$$

Then the formula ϕ is translated into the equation $\phi^* = y_{\mathrm{t}}$.

We recall here this simple logical fact that will be used in Lemma 8.1.9 and in Proposition 8.1.10.

Remark 8.1.8. Given an open formula $\phi \equiv \psi \vee \varphi$, we have that for any structure $\mathcal{M}$ and any environment $\rho : \mathrm{Var} \to M$, $\mathcal{M} \models_\rho \psi \vee \varphi$ iff $\mathcal{M} \models_\rho \psi$ or $\mathcal{M} \models_\rho \varphi$.

The next lemma shows that, given an interpretation of its free variables in Var, any term ϕ^* behaves similarly to a factor term.

Lemma 8.1.9. *Let $\mathbf{A}$ be a factor algebra, ϕ be an open formula and $\rho : \mathrm{Var} \to A$ be an environment. Then we have that*

$$\mathbf{A} \models_\rho \forall y_{\mathrm{t}} y_{\mathrm{f}}.\phi^* = y_{\mathrm{t}} \ \underline{\mathit{or}}\ \mathbf{A} \models_\rho \forall y_{\mathrm{t}} y_{\mathrm{f}}.\phi^* = y_{\mathrm{f}}. \tag{8.1}$$

Proof. The proof is by induction on the complexity of ϕ. We sometimes use the *abus de notation* $\mathbf{A} \models \forall y.\phi = t = q$, that means $\mathbf{A} \models \forall y.\phi = t \Leftrightarrow \mathbf{A} \models \forall y.t = q$.

If ϕ is y_{t} or y_{f}, then there is nothing to prove.

If $\phi \equiv R(\bar{t})$, then $\phi^* = \hat{R}(\bar{t}, y_{\mathrm{t}}, y_{\mathrm{f}})$. $\mathbf{A}$ is a factor algebra, so for any assignment of elements of A to the tuple t we have that $\forall xy.\hat{R}(\bar{t}, x, y) = x$ or $\forall xy.\hat{R}(\bar{t}, x, y) = y$ holds. The conclusion easily follows.

If $\phi \equiv \neg\psi$, then by ind. hyp. $\mathbf{A} \models_\rho \forall y_{\mathrm{t}} y_{\mathrm{f}}.\psi^* = y_{\mathrm{t}}$ or $\mathbf{A} \models_\rho \forall y_{\mathrm{t}} y_{\mathrm{f}}.\psi^* = y_{\mathrm{f}}$.

If $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\psi^* = y_\mathrm{t}$ holds, then we have $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\phi^* = \forall y_\mathrm{t} y_\mathrm{f}.\psi^*[y_\mathrm{t}/y_\mathrm{f}; y_\mathrm{f}/y_\mathrm{t}] = y_\mathrm{f}$.
If $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\psi^* = y_\mathrm{f}$ holds, then we have $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\phi^* = \forall y_\mathrm{t} y_\mathrm{f}.\psi^*[y_\mathrm{t}/y_\mathrm{f}; y_\mathrm{f}/y_\mathrm{t}] = y_\mathrm{t}$.

If $\phi \equiv \psi_1 \vee \psi_2$, by ind. hyp. condition 8.1 holds for ψ_1 and ψ_2.
If $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\psi_2^* = y_\mathrm{t}$, then $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\phi^* = \psi_2^*[\psi_1^*/y_\mathrm{f}] = \psi_2^* = y_\mathrm{f}$.
If $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\psi_2^* = y_\mathrm{f}$ and $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\psi_1^* = y_\mathrm{t}$, then $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\phi^* = \psi_2^*[\psi_1^*/y_\mathrm{f}] = \psi_1^* = y_\mathrm{t}$.
If $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\psi_2^* = y_\mathrm{f}$ and $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\psi_1^* = y_\mathrm{f}$, then $\mathbf{A} \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\phi^* = \psi_2^*[\psi_1^*/y_\mathrm{f}] = \psi_1^* = y_\mathrm{f}$.

The case $\phi \equiv \psi_1 \wedge \psi_2$ is similar to the previous one. □

We prove now the second result of semantical equivalence.

Proposition 8.1.10. *Let $\mathcal{M}$ be a structure, $\rho : \mathrm{Var} \to M$ be an environment and ϕ be an open propositional formula. Then,*

$$\mathcal{M} \models_\rho \phi \;\Leftrightarrow\; \mathrm{Fa}(\mathcal{M}) \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\phi^* = y_\mathrm{t}.$$

Proof. Proof is by induction on the complexity of ϕ.

$\mathcal{M} \models_\rho R(\bar{t})$	$\Leftrightarrow$	$\mathrm{Fa}(\mathcal{M}) \models_\rho \hat{R}(\bar{t}, y_t, y_f) = y_\mathrm{t}$	since $\mathrm{Fa}(\mathcal{M})$ is a non-trivial factor algebra.
$\mathcal{M} \models_\rho \neg\phi$	$\Leftrightarrow$	$\neg(\mathcal{M} \models_\rho \phi)$	
	$\Leftrightarrow$	$\mathrm{Fa}(\mathcal{M}) \not\models_\rho \phi^* = y_\mathrm{t}$	by ind. hyp.
	$\Leftrightarrow$	$\mathrm{Fa}(\mathcal{M}) \models_\rho \phi^* = y_\mathrm{f}$	by Lemma 8.1.9.
	$\Leftrightarrow$	$\mathrm{Fa}(\mathcal{M}) \models_\rho (\neg\phi)^* = \phi^*[y_\mathrm{t}/y_\mathrm{f}, y_\mathrm{f}/y_\mathrm{t}] = y_\mathrm{t}$	
$\mathcal{M} \models_\rho \phi_1 \vee \phi_2$	$\Leftrightarrow$	$\mathcal{M} \models_\rho \phi_1$ or $\mathcal{M} \models_\rho \phi_2$	
	$\Leftrightarrow$	$\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_1^* = y_\mathrm{t}$ or $\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_2^* = y_\mathrm{t}$	by ind. hyp.
	$\Leftrightarrow$	$\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_2^*[\phi_1^*/y_\mathrm{f}] = y_\mathrm{t}$	

We prove the last equivalence:

"$\Rightarrow$" If $\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_2^* = y_\mathrm{t}$ then $\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_2^*[\phi_1^*/y_\mathrm{f}] = y_\mathrm{t}[\phi_1^*/y_\mathrm{f}] = y_\mathrm{t}$.
If $\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_1^* = y_\mathrm{t}$ and $\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_2^* = y_\mathrm{f}$, then $\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_2^*[\phi_1^*/y_\mathrm{f}] = y_\mathrm{f}[\phi_1^*/y_\mathrm{f}] = \phi_1^* = y_\mathrm{t}$.

"$\Leftarrow$" Assume that $\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_1^* = y_\mathrm{f}$ and $\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_2^* = y_\mathrm{f}$. Then $\mathrm{Fa}(\mathcal{M}) \models_\rho \phi_2^*[\phi_1^*/y_\mathrm{f}] = y_\mathrm{f}[\phi_1^*/y_\mathrm{f}] = \phi_1^* = y_\mathrm{f}$: absurd. □

If $\mathcal{M}$ is a singleton structure, we only have $\mathcal{M} \models_\rho \phi \;\Rightarrow\; \mathrm{Fa}(\mathcal{M}) \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\phi^* = y_\mathrm{t}$. An example of a singleton structure in which "$\Leftarrow$" does not hold is given by a structure $\mathcal{M} = \{a\}$ on a type consisting of an unary relation R, interpreted as false in $\mathcal{M}$. We have $\mathcal{M} \not\models R(a)$, but $\mathrm{Fa}(\mathcal{M}) \models \hat{R}(a, a, a) = a$ and consequently $\mathrm{Fa}(\mathcal{M}) \models_\rho \forall y_\mathrm{t} y_\mathrm{f}.\hat{R}(x, y_\mathrm{t}, y_\mathrm{f}) = y_\mathrm{t}$.

8.1.4 Congruences associated with formulas

As a corollary of the translations developed in Section 8.1.3, we prove that an open formula ϕ of type τ canonically defines a pair of factor congruences on the algebras of a τ-factor variety V.

Definition 8.1.11. *Let V be a τ-factor variety, $\mathbf{A} \in \mathsf{V}$, $\rho : \mathrm{Var} \to A$ be an environment and ϕ be an open formula of type τ. We define:*

$$\theta_\phi = \{(c,d) : \mathbf{A} \models_\rho \phi^*[c/y_{\mathrm{t}}, d/y_{\mathrm{f}}] = c\}; \quad \theta_{\neg\phi} = \{(c,d) : \mathbf{A} \models_\rho (\neg\phi)^*[c/y_{\mathrm{t}}, d/y_{\mathrm{f}}] = c\}.$$

Notice that $\theta_{\neg\phi} = \{(c,d) : \mathbf{A} \models_\rho \phi^*[c/y_{\mathrm{t}}, d/y_{\mathrm{f}}] = d\}$.

Lemma 8.1.12. *If ϕ is logically equivalent to ψ under the environment ρ in $\mathbf{A}$, then $\theta_\phi = \theta_\psi$.*

Proof. If $\mathbf{A}$ is trivial, then there is nothing to prove. Otherwise, thanks to Proposition 8.1.6, we have that $\mathbf{A} \models_\rho \phi^* = y_{\mathrm{t}}$ iff $\mathbf{A} \models_\rho \psi^* = y_{\mathrm{t}}$. It follows that:

$$\begin{array}{rcl}
(c,d) \in \theta_\phi & \Leftrightarrow & \mathbf{A} \models_\rho \phi^*[c,d] = c \\
& \Leftrightarrow & \mathrm{Str}(\mathbf{A}) \models_\rho \phi \\
& \Leftrightarrow & \mathrm{Str}(\mathbf{A}) \models_\rho \psi \\
& \Leftrightarrow & \mathbf{A} \models_\rho \psi^*[c,d] = c \\
& \Leftrightarrow & (c,d) \in \theta_\psi
\end{array}$$

□

Proposition 8.1.13. *Let V be a factor variety, $\mathbf{A} \in \mathsf{V}$, $\rho : \mathrm{Var} \to A$ be an environment and ϕ be an open formula. Then $(\phi^*(-,-))^{\mathbf{A}}_\rho$ is a decomposition operator on $\mathbf{A}$.*

Proof. Proof is by induction on the complexity of ϕ.

If $\phi \equiv R(\bar{a})$, then $\hat{R}(\bar{a},-,-)$ is a decomposition operator because $\mathbf{A}$ satisfies equations F1, F2 and F3.

If $\phi = \neg\psi$, let $\psi^*(-,-)$ be the decomposition operator corresponding to the formula ψ. By Definition 8.1.7, the operator $\phi^*(-,-)$ satisfies the equation $\phi^*(x,y) = \psi^*(y,x)$. In order to simplify the notation, we denote $\psi^*(-,-)$ by $f(-,-)$, so $\phi^*(x,y) = f(y,x)$.

(D1) $\phi^*(x,x) = f(x,x) = x$.

(D2) $\phi^*(\phi^*(x_{11},x_{12}), \phi^*(x_{21},x_{22})) = f(f(x_{22},x_{21}), f(x_{12},x_{11})) = f(x_{22},x_{11}) = \phi^*(x_{11},x_{22})$.

(D3) Given any symbol $h \in \nu_\tau$ of arity n we have

$$\begin{array}{rcll}
\phi^*(h(x_1,\dots,x_n), h(y_1,\dots,y_n)) & = & f(h(y_1,\dots,y_n), h(x_1,\dots,x_n)) & \\
& = & h(f(y_1,x_1),\dots,f(y_n,x_n)) & \text{by ind. hyp.} \\
& = & h(\phi^*(x_1,y_1)\dots,\phi^*(x_n,y_n))) &
\end{array}$$

If $\phi \equiv \psi_1 \vee \psi_2$, let $\psi_i^*(-,-)$, $i = 1,2$ be decomposition operators corresponding to ψ_1, ψ_2 respectively. By Definition 8.1.7, $\phi^*(-,-)$ satisfies the equation $\phi^*(x,y) = \phi_2^*(x, \phi_1^*(x,y))$. We denote $\phi_2^*(-,-)$ by $f(-,-)$ and $\phi_1^*(-,-)$ by $g(-,-)$, so we can write $\phi^*(x,y) = f(x, g(x,y))$.

(D1) $\phi^*(x,x) = f(x, g(x,x)) = f(x,x) = x$.

(D2) We have that $\phi^*(\phi^*(x_{11}, x_{12}, \phi^*(x_{21}, x_{22})))$ is equal to

$$f(f(x_{11}, g(x_{11}, x_{12})), g(f(x_{11}, g(x_{11}, x_{12})), g(f(x_{21}, g(x_{21}, x_{22}))))) \tag{8.2}$$

Now we consider the subterm $g(f(x_{11}, g(x_{11}, x_{12})), f(x_{21}, g(x_{21}, x_{22})))$ 8.2. It can be reduced as follows:

$$\begin{array}{lcl} g(f(x_{11}, g(x_{11}, x_{12})), f(x_{21}, g(x_{21}, x_{22}))) & = & \text{by ind. hyp. on } g \\ f(g(x_{11}, x_{21}), g(g(x_{11}, x_{12}), g(x_{21}, x_{22}))) & = & \text{by ind. hyp. on } g \\ f(g(x_{11}, x_{21}), g(x_{11}, x_{22})) & & \end{array}$$

So 8.2 can be simplified as follows:

$$\begin{array}{lcl} f(f(x_{11}, g(x_{11}, x_{12})), f(g(x_{11}, x_{21}), g(x_{11}, x_{22}))) & = & \text{by ind. hyp. on } f \\ f(x_{11}, g(x_{11}, x_{22})) & = & \\ \phi^*(x_{11}, x_{22}). & & \end{array}$$

(D3) Given any symbol $h \in \nu_\tau$ of arity n. we have that

$$\begin{array}{lcl} \phi^*(h(x_1, \dots, x_n), h(y_1, \dots, y_n)) & = & \\ f(h(x_1, \dots, x_n), g(h(x_1, \dots, x_n), h(y_1, \dots, y_n)) & = & \text{by ind. hyp. on } g \\ f(h(x_1, \dots, x_n), h(g(x_1, y_1), \dots, g(x_n, y_n))) & = & \text{by ind. hyp. on } f \\ h(f(x_1, g(x_1, y_1)), \dots, f(x_n, g(x_n, y_n))) & = & \\ h(\phi^*(x_1, y_1), \dots, \phi^*(x_n, y_n)) & & \end{array}$$

The case $\phi_1 \wedge \phi_2$ is very similar to the previous one.

□

Corollary 8.1.14. *Under the hypotheses of Proposition 8.1.13, $(\theta_\phi, \theta_{\neg\phi})$ is a pair of complementary factor congruences.*

8.2 A new calculus for classical logic

In this section we use factor algebras of first-order types to develop an algebraic calculus for classical propositional logic. It is based on the correspondence between logical and algebraic notions that we have developed in the previous sections.

We fix a type τ_{CL} in which it is possible to express any propositional formula and then we build the factor variety $\mathsf{Fa}_{\tau_{CL}}$. Then we show that the axioms defining $\mathsf{Fa}_{\tau_{CL}}$

are the rules for the calculus: indeed, in Theorem 8.2.4 we prove that a propositional formula ϕ is a tautology iff $\mathsf{Fa}_{\tau_{CL}}$ satisfies the equation $\phi^* = y_{\mathrm{t}}$.

Let $\tau_{CL} = (\varnothing; A_i)_{i\in\omega}$ be a relational type where A_i are all propositional variables (i.e., A_i has arity 0 for every i).

Definition 8.2.1. *The variety* $\mathsf{Fa}_{\tau_{CL}}$ *of type* $\nu_{\tau_{CL}}$ *is axiomatized by the following axioms.*

For all $i, k \in \omega$*:*

(G1) *Idempotence:* $\hat{A}_i(x, x) = x$.

(G2) *Associativity:* $\hat{A}_i(\hat{A}_i(x, y), z) = \hat{A}_i(x, \hat{A}_i(y, z))$.

(G3) *Distributivity:*

$$\hat{A}_i(x, \hat{A}_k(y, z)) = \hat{A}_k(\hat{A}_i(x, y), \hat{A}_i(x, z)); \quad \hat{A}_i(\hat{A}_k(x, y), z) = \hat{A}_k(\hat{A}_i(x, z)\hat{A}_i(y, z)).$$

(G4) *Collapsing:* $\hat{A}_i(x, \hat{A}_i(y, z)) = \hat{A}_i(x, z)$.

Sometimes the above notation is not convenient. So in the following proposition and in some examples we denote expressions of the form $\hat{A}_i(x, y)$ by $x \cdot_i y$.

Proposition 8.2.2. $\mathsf{Fa}_{\tau_{CL}}$ *is a* τ_{CL}*-factor variety.*

Proof. We show that the axioms of Definition 8.2.1 prove the equations of Theorem 7.3.3.

(F1) $x \cdot_i x =_{G1} x$.

(F2) $(x \cdot_i y) \cdot_i (z \cdot_i v) =_{G2} x \cdot_i y \cdot_i z \cdot_i v =_{G4} x \cdot_i v$.

(F3) $(x \cdot_i y) \cdot_k (z \cdot_i v) =_{G3} ((x \cdot_i y) \cdot_k z) \cdot_i ((x \cdot_i y) \cdot_k z) =_{G3} ((x \cdot_k z) \cdot_i (y \cdot_k z)) \cdot_i ((x \cdot_k v) \cdot_i (y \cdot_k v)) =_{G4} (x \cdot_k z) \cdot_i (y \cdot_k v)$.

□

It is straightforward to check that for every $\mathcal{M} \in \mathbb{M}_{\tau_{CL}}$, where $\mathbb{M}_{\tau_{CL}}$ is the class of all structures of type τ_{CL}, $\mathrm{Fa}(\mathcal{M})$ is a factor algebra in the variety $\mathsf{Fa}_{\tau_{CL}}$. Conversely, a simple calculation shows that every τ_{CL}-factor algebra satisfies axioms G1,G2, G3 and G4.

So we have the following identification.

Proposition 8.2.3. *The variety generated by* $\{\mathrm{Fa}(\mathcal{M}) : \mathcal{M} \in \mathbb{M}_{\tau_{CL}}\}$ *is equal to* $\mathsf{Fa}_{\tau_{CL}}$.

Thanks to Proposition 8.2.3, we prove one of the main results of the chapter.

Theorem 8.2.4. *Let ϕ be a propositional formula.*
ϕ is a tautology iff for every algebra $\mathbf{A} \in \mathsf{Fa}_{\tau_{CL}}$, $\mathbf{A} \models \forall y_{\mathrm{t}} y_{\mathrm{f}}.\phi^* = y_{\mathrm{t}}$

Proof. By Proposition 8.1.10 we have that for any structure $\mathcal{M}$ of type τ_{CL},

$$\mathcal{M} \models \phi \quad \text{iff} \quad \mathrm{Fa}(\mathcal{M}) \models \forall y_{\mathrm{t}} y_{\mathrm{f}}.\phi^* = y_{\mathrm{t}}.$$

By Proposition 8.2.3 we can conclude that

$$\phi \text{ is a tautology} \quad \text{iff} \quad \mathbf{A} \models \forall y_{\mathrm{t}} y_{\mathrm{f}}.\phi^* = y_{\mathrm{t}} \text{ for every } \mathbf{A} \in \mathsf{Fa}_{\tau_{CL}}.$$

□

Example 8.2.5. In this example we denote $A_i(x, y)$ by $x \cdot y$.

We prove that $\phi = A \vee \neg A$ is a tautology. Since $\phi^* = y_{\mathrm{t}} \cdot (y_{\mathrm{f}} \cdot y_{\mathrm{t}})$ then we can reduce as follows:

$$\begin{aligned} y_{\mathrm{t}} \cdot (y_{\mathrm{f}} \cdot y_{\mathrm{t}}) &= y_{\mathrm{t}} \cdot y_{\mathrm{f}} \cdot y_{\mathrm{t}} && \text{by G2} \\ &= y_{\mathrm{t}} \cdot y_{\mathrm{t}} && \text{by G4} \\ &= y_{\mathrm{t}} && \text{by G1} \end{aligned}$$

Example 8.2.6. In this example we consider the Peirce Law $\psi = ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$. In order to apply Definition 8.1.7 we put ψ in the form $((\neg A \vee B) \wedge \neg A) \vee A$. We have

$$\psi^* = \hat{B}(\hat{A}(\hat{A}(y_{\mathrm{t}}, y_{\mathrm{f}}), y_{\mathrm{t}}), \hat{A}(\hat{A}(y_{\mathrm{t}}, y_{\mathrm{f}}), \hat{A}(\hat{A}(y_{\mathrm{t}}, y_{\mathrm{f}}), y_{\mathrm{t}})))$$

that corresponds to the expression

$$((y_{\mathrm{t}} \cdot_A y_{\mathrm{f}}) \cdot_A y_{\mathrm{t}}) \cdot_B ((y_{\mathrm{t}} \cdot_A y_{\mathrm{f}}) \cdot_A ((y_{\mathrm{t}} \cdot_A y_{\mathrm{f}}) \cdot_A y_{\mathrm{t}})).$$

It can be reduced as follows:

$$\begin{aligned} ((y_{\mathrm{t}} \cdot_A y_{\mathrm{f}}) \cdot_A y_{\mathrm{t}}) \cdot_B ((y_{\mathrm{t}} \cdot y_{\mathrm{f}}) \cdot_A ((y_{\mathrm{t}} \cdot_A y_{\mathrm{f}}) \cdot_A y_{\mathrm{t}})) &= (y_{\mathrm{t}} \cdot_A y_{\mathrm{t}}) \cdot_B (y_{\mathrm{t}} \cdot_A y_{\mathrm{t}}) && \text{by } \mathrm{G4_A} \\ & && \text{and } \mathrm{G2_A} \\ &= y_{\mathrm{t}} \cdot_B y_{\mathrm{t}} && \text{by } \mathrm{G1_A} \\ &= y_{\mathrm{t}} && \text{by } \mathrm{G1_B} \end{aligned}$$

Example 8.2.7. If we consider the formula $\phi = A \vee A$, we have $\phi^* = \hat{A}(y_{\mathrm{t}}, \hat{A}(y_{\mathrm{t}}, y_{\mathrm{f}}))$: it is algebraically interpreted as $y_{\mathrm{t}} \cdot_A y_{\mathrm{t}} \cdot_A y_{\mathrm{f}}$, and it can be reduced only to $y_{\mathrm{t}} \cdot_A y_{\mathrm{f}}$, because $A \vee A$ is not a tautology.

8.2.1 A confluent and terminating strategy for the calculus

This section was mainly developed by Giulio Manzonetto.

The formulas of the examples in the previous section are very simple, so computations were easy to handle: when a formula has many propositional variables, in general it is not clear how to use effectively axiom (G3).

In this section we define a terminating and confluent term rewriting system (TRS for short) for the calculus. This system can be used as a concrete algorithm to check whether a formula is a tautology or not.

Given a type τ, $\mathbf{T}(\tau, \mathrm{Var})$ denotes the set of τ-terms built on the set Var of variables.

Definition 8.2.8. *A TRS (τ, R) is a set of rewriting rules R on the set of terms of type τ.*

8.2.1.1 Reduction rules

Let $\tau_K = \{y_{\mathrm{t}}, y_{\mathrm{f}}\} \cup \{A_i : i \in \omega\}$ be a type where $y_{\mathrm{t}}, y_{\mathrm{f}}$ are constants and the A_i's are binary operators. In the following functions $\hat{A}_i(-,-)$ are denoted as $A_i(-,-)$.

Definition 8.2.9. *The TRS $(\tau_K, \mathcal{R})$ is defined on $\mathbf{T}(\tau_K, \mathrm{Var})$ according to the following rules $\mathcal{R}$.*

For all $i, k \in \omega$:

(D1) $A_i(x,x) \to x$.

(D2a) $A_i(A_i(x,y), A_i(u,z)) \to A_i(x,z)$.

(D2b) $A_i(x, A_i(y,z)) \to A_i(x,z)$.

(D2c) $A_i(A_i(x,y), z) \to A_i(x,z)$.

(D3a) *If $i > k$, then* $A_i(A_k(x,y), A_k(z,u)) \to A_k(A_i(x,z), A_i(y,u))$.

(D3b) *If $i > k$, then* $A_i(x, A_k(y,z)) \to A_k(A_i(x,y), A_i(x,z))$.

(D3c) *If $i > k$, then* $A_i(A_k(x,y), z) \to A_k(A_i(x,z), A_i(y,z))$.

In order to prove the termination of TRS $(\tau_K, \mathcal{R})$ we need to recall some results of the theory of rewriting systems.

Definition 8.2.10.
1. *A* rewrite order *is a partial order $(\mathbf{T}(\tau, \mathrm{Var}), >)$ which is closed under contexts (i.e., if $s > t$ then $C[s] > C[t]$ for all contexts $C[\ ]$) and closed under substitutions (i.e. if $s > t$ then $s\sigma > t\sigma$ for all substitutions σ).*

2. *A* simplification order $(\mathbf{T}(\tau, \mathrm{Var}), >)$ *is a rewrite order having the subterm property, i.e., $C[t] > t$ for all contexts $C[\ ] \neq [\ ]$.*

3. *A TRS (τ, R) is* simplifying *if it is compatible with a simplification order $(\mathbf{T}(\tau, \mathrm{Var}), >)$, i.e., for any rule $\ell \to r \in R$, we have $\ell > r$.*

Our goal is to define a simplification order compatible with the TRS $(\tau_K, \mathcal{R})$.

First we endow τ_K with a strict order $>$, induced by the indices:

- $A_i > y_{\mathrm{t}} > y_{\mathrm{f}}$ for all $i \in \omega$,
- $A_i > A_j \iff i > j$.

By a result described in [42] , a strict order $>$ on a type τ induces a (unique) lexicographic path order (LPO) on $\mathbf{T}(\tau_K, \mathrm{Var})$.

Definition 8.2.11. *The lexicographic path order $>_{\mathrm{lpo}}$ on $\mathbf{T}(\tau_K, \mathrm{Var})$ is defined as follows: $s >_{\mathrm{lpo}} t$ iff*

(LPO1) *$t \in Var(s)$ and $s \neq t$, or*

(LPO2) *$s = A_i(s_1, s_2)$, $t = A_j(t_1, t_2)$ and one of the following conditions holds:*

> **(LPO2a)** $(\exists k \in [1, 2])(s_k \geq_{\mathrm{lpo}} t)$,
> **(LPO2b)** *$i > j$, and* $(\forall k \in [1, 2])(s >_{\mathrm{lpo}} t_k)$,
> **(LPO2c)** *$i = j$, $\langle s_1, s_2\rangle >^{\mathrm{lex}}_{\mathrm{lpo}} \langle t_1, t_2\rangle$, and* $(\forall k \in [1, 2])(s >_{\mathrm{lpo}} t_k)$.

The general definition of lexicographic path order can be found in [29, 0.23].

Lemma 8.2.12. *[64, Proposition 6.4.25] Every lexicographic partial order is a simplification order.*

Proposition 8.2.13. *The TRS $(\tau_K, \mathcal{R})$ is simplifying because it is compatible with $>_{\mathrm{lpo}}$.*

Proof. In the following we suppose $i > j$.

(D1) $A_i(x, x) >_{\mathrm{lpo}} x$ by LPO1.

(D2a) $A_i(A_i(x, y), A_i(u, z)) >_{\mathrm{lpo}} A_i(x, z)$ by LPO2c, since we have:

- $\langle A_i(x, y), A_i(u, z)\rangle >^{\mathrm{lex}}_{\mathrm{lpo}} \langle x, z\rangle$ as $A_i(x, y) >_{\mathrm{lpo}} x$ by LPO1,
- $A_i(A_i(x, y), A_i(u, z)) >_{\mathrm{lpo}} x$ and $A_i(A_i(x, y), A_i(u, z)) >_{\mathrm{lpo}} z$ by LPO1.

(D2b) $A_i(x, A_i(y, z)) >_{\mathrm{lpo}} A_i(x, z)$ by LPO2c, since we have:

- $\langle x, A_i(y, z)\rangle >^{\mathrm{lex}}_{\mathrm{lpo}} \langle x, z\rangle$ as the first components coincide and $A_i(y, z) >_{\mathrm{lpo}} z$ by LPO1,
- $A_i(x, A_i(y, z)) >_{\mathrm{lpo}} x$ and $A_i(x, A_i(y, z)) >_{\mathrm{lpo}} z$ by LPO1.

(D2c) $A_i(A_i(x, y), z) >_{\mathrm{lpo}} A_i(x, z)$ by LPO2c since:

- $\langle A_i(x, y), z\rangle >^{\mathrm{lex}}_{\mathrm{lpo}} \langle x, z\rangle$ as $A_i(x, y) >_{\mathrm{lpo}} x$ by LPO1,
- $A_i(A_i(x, y), z) >_{\mathrm{lpo}} x$ and $A_i(A_i(x, y), z) >_{\mathrm{lpo}} z$ by LPO1.

(D3a) $A_i(A_j(x, y), A_j(z, u)) >_{\mathrm{lpo}} A_j(A_i(x, z), A_i(y, u))$ by LPO2b since $i > j$ and

- $A_i(A_j(x,y), A_j(z,u)) >_{\mathrm{lpo}} A_i(x,z)$ by LPO2c. Indeed we have, by LPO1, $\langle A_j(x,y), A_j(z,u)\rangle >^{\mathrm{lex}}_{\mathrm{lpo}} \langle x,z\rangle$ since $A_j(x,y) >_{\mathrm{lpo}} x$, and, still by LPO1, $A_i(A_j(x,y), A_j(z,u)) >_{\mathrm{lpo}} x$ and $A_i(A_j(x,y), A_j(z,u)) >_{\mathrm{lpo}} z$.
- $A_i(A_j(x,y), A_j(z,u)) >_{\mathrm{lpo}} A_i(y,u)$ by LPO2c. Indeed we have, by LPO1, $\langle A_j(x,y), A_j(z,u)\rangle >^{\mathrm{lex}}_{\mathrm{lpo}} \langle y,u\rangle$ since $A_j(x,y) >_{\mathrm{lpo}} y$, and, still by LPO1, $A_i(A_j(x,y), A_j(z,u)) >_{\mathrm{lpo}} y$ and $A_i(A_j(x,y), A_j(z,u)) >_{\mathrm{lpo}} u$.

(D3b) $A_i(x, A_j(y,z)) \to A_j(A_i(x,y), A_i(x,z))$ by LPO2b since $i > j$ and

- $A_i(x, A_j(y,z)) >_{\mathrm{lpo}} A_i(x,y)$ by LPO2c. Indeed $\langle x, A_j(y,z)\rangle >^{\mathrm{lex}}_{\mathrm{lpo}} \langle x,y\rangle$ since $A_j(y,z) >_{\mathrm{lpo}} y$ by LPO1, moreover $A_i(x, A_j(y,z)) >_{\mathrm{lpo}} x$ and $A_i(x, A_j(y,z)) >_{\mathrm{lpo}} y$ again by LPO1.
- $A_i(x, A_j(y,z)) >_{\mathrm{lpo}} A_i(x,z)$ by LPO2c. Indeed $\langle x, A_j(y,z)\rangle >^{\mathrm{lex}}_{\mathrm{lpo}} \langle x,z\rangle$ since the first components coincide and $A_j(y,z) >_{\mathrm{lpo}} z$ by LPO1, moreover $A_i(x, A_j(y,z)) >_{\mathrm{lpo}} x$ and $A_i(x, A_j(y,z)) >_{\mathrm{lpo}} z$ again by LPO1.

(D3c) $A_i(A_j(x,y), z) \to A_j(A_i(x,z), A_i(y,z))$ similar to the previous case.

□

8.2.1.2 Termination

Given a term t we write $Fun(t)$ for the set of function symbols occurring in t.

Definition 8.2.14. *Given a TRS (τ, R), we set*

$$\mathcal{F} = \bigcup_{\ell \to r \in R} (Fun(r) \setminus Fun(\ell))$$

i.e. $\mathcal{F}$ consists of all those function symbols which occur at the right-hand side r but not at the left-hand side ℓ of some rule $\ell \to r \in R$. We say that the TRS (τ, R) introduces only finitely many function symbols *if the set $\mathcal{F}$ is finite.*

To prove that the TRS $(\tau_K, \mathcal{R})$ is terminating we need the following result.

Theorem 8.2.15. *([55, Theorem 4.13]) A simplifying TRS (τ, R) that introduces only finitely many function symbols is terminating.*

Theorem 8.2.16. *The TRS $(\tau_K, \mathcal{R})$ is terminating.*

Proof. By Lemma 8.2.12, $>_{\mathrm{lpo}}$ is a simplification order and by Proposition 8.2.13, the TRS $(\tau_K, \mathcal{R})$ is simplifying. Since the rules of the TRS do not introduce any new function symbol we have $\mathcal{F} = \varnothing$: by Theorem 8.2.15 we conclude that it is terminating. □

8.2.1.3 Confluence

For confluence we have to check that all critical pairs can be reduced to the same term.

Lemma 8.2.17. *The TRS $(\tau_K, \mathcal{R})$ is locally confluent.*

Proof. Let $i, k \in \omega$ such that $i > j$.

- Since $A_i(A_i(x,x), A_i(u,z)) \to_{D2a} A_i(x,z)$ and $A_i(A_i(x,x), A_i(u,z)) \to_{D1} A_i(x, A_i(u,z))$, we have the critical pair $\langle A_i(x,z), A_i(x, A_i(u,z))\rangle$, that can be joined as $A_i(x, A_i(u,z)) \to_{D2b} A_i(x,z)$.
- Since $A_i(A_i(x,y), A_i(z,z)) \to_{D2a} A_i(x,z)$ and $A_i(A_i(x,y), A_i(z,z)) \to_{D1} A_i(A_i(x,y), z)$, we have the critical pair $\langle A_i(x,z), A_i(A_i(x,y),z)\rangle$, that can be joined as $A_i(A_i(x,y),z) \to_{D2c} A_i(x,z)$.
- Since we have $A_i(A_j(x,x), A_j(z,u)) \to_{D3a} A_j(A_i(x,z), A_i(x,u))$ and $A_i(A_j(x,x), A_j(z,u)) \to_{D1} A_i(x, A_j(z,u))$, we must join the critical pair $\langle A_j(A_i(x,z), A_i(x,u)), A_i(x, A_j(z,u))\rangle$: this can be done as $A_i(x, A_j(z,u)) \to_{D3b} A_j(A_i(x,z), A_i(x,u))$.
- Since we have $A_i(A_j(x,y), A_j(z,z)) \to_{D3a} A_j(A_i(x,z), A_i(y,z))$ and $A_i(A_j(x,y), A_j(z,z)) \to_{D1} A_i(A_j(x,y), z)$, we must join the critical pair $\langle A_j(A_i(x,z), A_i(y,z)), A_i(A_j(x,y),z)\rangle$: this can be done as $A_i(A_j(x,y),z) \to_{D3c} A_j(A_i(x,z), A_i(y,z))$.
- Since we have $A_i(x, A_j(z,z)) \to_{D3b} A_j(A_i(x,z), A_i(x,z))$ and $A_i(x, A_j(z,z)) \to_{D1} A_i(x,z)$ we must join the critical pair $\langle A_j(A_i(x,z), A_i(x,z)), A_i(x,z)\rangle$: this can be done as $A_j(A_i(x,z), A_i(x,z)) \to_{D2a} A_i(x,z)$.
- Since $A_i(A_j(x,x), z) \to_{D3c} A_j(A_i(x,z), A_i(x,z))$ and $A_i(A_j(x,x), z) \to_{D1} A_i(x,z)$ we have the critical pair $\langle A_j(A_i(x,z), A_i(x,z)), A_i(x,z)\rangle$, that can be joined as $A_j(A_i(x,z), A_i(x,z)) \to_{D2c} A_i(x,z)$.
- Since $A_i(A_j(x,y), A_i(u,z)) \to_{D2b} A_i(A_j(x,y), z)$ and $A_i(A_j(x,y), A_i(u,z)) \to_{D3c} A_j(A_i(x, A_i(z,u)), A_i(y, A_i(z,u)))$ we have the critical pair
$$\langle A_i(A_j(x,y), z), A_j(A_i(x, A_i(z,u)), A_i(y, A_i(z,u)))\rangle,$$
that can be joined as
$$\begin{array}{c} A_i(A_j(x,y),z)) \\ \downarrow^{D3c} \\ A_j(A_i(x,u), A_i(y,u)) \\ \uparrow^{2}_{D2b} \\ A_j(A_i(x, A_i(z,u)), A_i(y, A_i(z,u))). \end{array}$$

- Since $A_i(A_i(x,y),A_j(u,z)) \rightarrow_{D2c} A_i(x,A_j(u,z))$ and $A_i(A_j(x,y),A_i(u,z))$ $\rightarrow_{D3b} A_j(A_i(A_i(x,y),u),A_i(A_i(x,y),z))$ we have the critical pair

$$\langle A_i(x,A_j(u,z)), A_j(A_i(A_i(x,y),u),A_i(A_i(x,y),z))\rangle$$

that can be joined as

$$\begin{array}{c} A_i(x,A_j(u,z)) \\ \downarrow^{D3b} \\ A_j(A_i(x,u),A_i(x,z)) \\ \uparrow^2_{D2c} \\ A_j(A_i(A_i(x,y),u),A_i(A_i(x,y),z)). \end{array}$$

- Let $i > j > q$. Since we have the following rewriting steps $A_i(A_j(x,y),A_q(z,u))$ $\rightarrow_{D3c} A_j(A_i(x,A_q(z,u)),A_i(y,A_q(z,u)))$ and $A_i(A_j(x,y),A_q(z,u)) \rightarrow_{D3b}$ $(A_i(A_j(x,y),z),A_i(A_j(x,y),u))$, we must join the critical pair:

$$\langle A_j(A_i(x,A_q(z,u)),A_i(y,A_q(z,u))), A_q(A_i(A_j(x,y),z),A_i(A_j(x,y),u))\rangle.$$

This can be done as

$$\begin{array}{c} A_j(A_i(x,A_q(z,u)),A_i(y,A_q(z,u))) \\ \downarrow^{D3b}_2 \\ A_j(A_q(A_i(x,z),A_i(x,u)),A_q(A_i(y,z),A_i(y,u)) \\ \downarrow^{D3a} \\ A_q(A_j(A_i(x,z),A_i(y,z)),A_j(A_i(x,u),A_i(y,u)) \\ \uparrow^2_{D3c} \\ A_q(A_i(A_j(x,y),z),A_i(A_j(x,y),u)) \end{array}$$

All other cases are of the form $r \leftarrow t \rightarrow r$ so they do not need any proof. □

Since $(\tau_K, \mathcal{R})$ is terminating (Theorem 8.2.16) and locally confluent we can apply Newman's lemma ([54]): if a TRS is terminating and locally confluent, then it is confluent.

Corollary 8.2.18. *The TRS $(\tau_K, \mathcal{R})$ is confluent.*

8.2.2 Algebraic structures and logical systems

In this section we show that well-known algebraic varieties can be interpreted as logics.

1. **Rectangular bands = classical logic with one propositional variable**. The variety of rectangular bands is the class of all groupies satisfying the following axioms:

- idempotence: $xx = x$.
- associativity: $(xy)z = x(yz)$.
- collapsing: $xyz = xz$.

Let $\tau_1 = (\varnothing; R)$ be a relational type with only one propositional variable R. In the following we write xy for $\hat{R}(x, y)$.

We show that FA_{τ_1} is the variety of rectangular bands.

(G1) is idempotence $xx = x$.

(G2) is associativity $x(yz) = (xy)z$.

(G4) is collapsing $xyz = xz$.

(G3) i.e., $xyzu = xzyu$, is a consequence of $xyz = xz$.

The factor algebras in the variety of rectangular bands are the left-zero bands and the right zero bands.

In this context a first-order structure is either a pair $\mathcal{M}_1 = (M, \text{R=true})$ or a pair $\mathcal{M}_0 = (M, \text{R=false})$. In the first case $\mathrm{Fa}(\mathcal{M})$ is a left-zero band, while in the second one a right-zero band.

2. **Distributive double rectangular bands = classical logic with two propositional variables**.

The variety of distributive rectangular bands is the class of algebras with two binary operations $\cdot$ (denoted by xy) and $+$ satisfying the following axioms:

- $(\mathbf{A}, \cdot)$ and $(\mathbf{A}, +)$ are both rectangular bands.
- Distributive laws:

$$x(y+z) = xy + xz; \qquad x + yz = (x+y)(x+z);$$

and

$$(y+z)x = yx + zx; \qquad yz + x = (y+x)(z+x).$$

Let $\tau_2 = (\varnothing; R, S)$ be a relational type with only two propositional variables R and S. In the following we write xy for $\hat{R}(x, y)$ and $x + y$ for $\hat{S}(x, y)$.

The variety of distributive rectangular bands is the variety FA_{τ_2}. The only non trivial task is to prove Axiom (G3). We have that $xy + zt = (x + zt)(y + zt) = (x+z)(x+t)(y+z)(y+t)$. Conversely, $x(y+z) = (x+x)(y+z) = xy+xz$ and similarly for the other laws. So (G3) is equivalent to the previous distributive laws.

We have four kinds of factor algebras in FA_{τ_2}:

(1) left-left-zero double bands satisfying $xy = x = x + y$;

(2) right-right-zero double bands satisfying $xy = y = x + y$;

(3) left-right-zero double bands satisfying $xy = x = y + x$;

(4) right-left-zero double bands satisfying $xy = y = y + x$.

In this context we have four kinds of first-order structures, which corresponds to the four kinds of factor algebras:

(1) $\mathcal{M}_{11} = (M, \text{R=true}, \text{S=true})$.

(2) $\mathcal{M}_{10} = (M, \text{R=true}, \text{S=false})$.

(3) $\mathcal{M}_{01} = (M, \text{R=false}, \text{S=true})$.

(4) $\mathcal{M}_{00} = (M, \text{R=false}, \text{S=false})$.

3. **Distributive ω-rectangular bands = classical logic with denumerable many propositional variables**. The variety of distributive rectangular bands is the class of algebras with denumerable many binary operations $\cdot_i$ satisfying the following axioms:

   - for any i, $(\mathbf{A}, \cdot_i)$ is a rectangular band.
   - Distributive laws: for any i, j,

   $$(x \cdot_j y) \cdot_i z = (x \cdot_i z) \cdot_j (y \cdot_i z); \qquad x \cdot_i (y \cdot_j z) = (x \cdot_i y) \cdot_j (x \cdot_i z).$$

   The variety of distributive rectangular bands is equal to $\mathsf{FA}_{\tau_{CL}}$.

4. **Rectangular Skew Lattices = classical logic with two propositional variables R, S such that $R \leftrightarrow \neg S$**. The variety of rectangular skew lattices is axiomatized by the equations defining distributive rectangular bands plus the axiom $x + y = yx$.

   It can be described as the subvariety of the variety of distributive double rectangular bands generated by the right-left-zero double bands and left-right-zero double bands is the variety of rectangular skew lattices.

5. **Discriminator Varieties = classical logic with equality (no other relation symbol)**. Let τ be a type whose unique relation symbol R is binary. Following Vaggione [67], FA_τ is a discriminator variety if, and only if, it satisfies, besides the axioms of factor variety, $\hat{R}(x, x, y, z) = y$ and $\hat{R}(x, y, x, y) = y$.

6. **Skew Boolean Algebras = classical logic with one unary relation symbol R satisfying $\neg\mathbf{R}(\mathbf{0}) \land \forall \mathbf{x}(\neg\mathbf{R}(\mathbf{x}) \to \mathbf{x} = \mathbf{0})$**. Let $\tau = (0; R)$ be a type of arity $(0; 1)$. Following Cvetko-Vah and Salibra [28], the factor variety axiomatized by $\hat{R}(0, y, z) = z$ and $\hat{R}(x, x, 0) = x$ (besides the axioms of factor variety) is term equivalent to the variety of skew Boolean algebras. A factor algebra $\mathbf{A} = (A, 0, \hat{R})$ in this variety satisfies $\hat{R}(0, y, z) = z$ and $\hat{R}(x, y, z) = y$ for all $x \in A \setminus \{0\}$.

7. **Boolean Algebras = classical logic with one unary relation symbol R satisfying $\neg\mathbf{R}(\mathbf{0}) \wedge \mathbf{R}(\mathbf{1}) \wedge \forall\mathbf{x}(\mathbf{x} = \mathbf{0} \vee \mathbf{x} = \mathbf{1})$**. Let $\tau = (0, 1; R)$ be a type of arity $(0, 0; 1)$. Following Salibra et al. [57], the factor variety axiomatized by $\hat{R}(0, y, z) = z$, $\hat{R}(1, y, z) = y$ and $\hat{R}(x, 1, 0) = x$ is term equivalent to the variety of Boolean algebras. Up to isomorphism, we have one factor algebra $\mathbf{A} = (\{0, 1\}, 0, 1, \hat{R})$ satisfying $\hat{R}(0, y, z) = z$ and $\hat{R}(1, y, z) = y$.

8. **Ordered Algebras = classical logic with one binary relation symbol defining a compatible partial ordering**. Let $\tau = \{R\}$ be a type of arity 2. We write $o(x, y, z, u)$ for $\hat{R}(x, y, z, u)$.

   FA_τ is a *factor variety of ordered algebras* if, and only if, it satisfies axioms F1-F2-F3 and the following identities:

   (O_1) $o(x, x, y_\mathrm{t}, y_\mathrm{f}) = y_\mathrm{t}$ (Reflexivity)

   (O_2) $o(x, z, y_\mathrm{t}, o(y, z, o(x, y, y_\mathrm{f}, y_\mathrm{t}), y_\mathrm{t})) = y_\mathrm{t}$ (Transitivity);

   (O_3) $o(x, y, x, y) = o(y, x, y, o(x, y, x, y))$ (Antisymmetry);

   (O_4) For every function symbol g of arity k,

   $$o(g(z_1, \dots, x, \dots, z_k), g(z_1, \dots, y, \dots, z_k), y_\mathrm{t}, o(x, y, y_\mathrm{f}, y_\mathrm{t})) = y_\mathrm{t}$$

   where x and y are in the i^{th} entry. (Monotonicity in coordinate i).

   Any ordered factor algebra $\mathbf{A}$ is simple. Let a, b different elements of $\mathbf{A}$. We show that (a, b) splits A. We have that $a \neq b$ implies $a \not\leq b$. Then we have

   $$o(a, a, x, y) = o(a, b, y, x) = x.$$

8.3 Applications

8.3.1 Factor varieties and symbolic computation

In this section Σ is a set of sentences and ϕ is a sentence on the same type.

MacKenzie (see Burris [22]) has shown that it is possible to routinely cast the semantical problem $\Sigma \models \phi$ as an equational problem in discriminator varieties. This interesting result uses a enough involved technique which is briefly explained below.

Given Σ, ϕ and a discriminator variety V, he introduces a reduction procedure that defines a set of equational axioms Ax. Then he proves that $\Sigma \models \phi$ if and only if (a) the singleton models of Σ are models of ϕ; (b) The equational axioms Ax prove $\forall xy.x = y$.

By using factor algebras it is possible to simplify some technical steps. In particular, steps 4 and 5 of page 197 of [22] are replaced by step 4 of 8.3.1.1.

8.3.1.1 Reduction to equations

We rephrase the reduction procedure by using factor varieties.

1. We put all sentences of $\Sigma \cup \{\neg\phi\}$ in prenex form.

2. We skolemize the sentences obtained in step 1.

3. We add to the type of Σ and ϕ the new function symbols. We denote the resulting type by τ'.

4. Consider the factor variety V generated by the following equations, valid for any relation symbol R in τ.

   (F1) $\hat{R}(\bar{x}, z, z) = z$;

   (F2) $\hat{R}(\bar{x}, \hat{R}(\bar{x}, x_{11}, x_{12}), \hat{R}(\bar{x}, x_{21}, x_{22})) = \hat{R}(\bar{x}, x_{11}, x_{22})$.

   (F3) $\hat{R}(\bar{x}, h(\bar{y}), h(\bar{z})) = h(\hat{R}(\bar{x}, y_1, z_1), \hat{R}(\bar{x}, y_1, z_1), \dots, \hat{R}(\bar{x}, y_k, z_k))$, where $h \in \nu_{\tau'}$ of arity k.

5. We replace the matrices of the universal formulas obtained in step 2 with equations $\psi^* = y_{\mathrm{t}}$ by using rules of Definition 8.1.7.

Remark 8.3.1. Since equality is in τ, V is a discriminator variety with some factor terms.

We denote by Γ the set of equations obtained in point 5 and 4.

Theorem 8.3.2. $\Sigma \models \phi$ *iff*

(i) every singleton model of Σ *is a model of* ϕ.

(ii) $\Gamma \models \forall xy.x = y$

Proof. By Theorem 8.1.10 , in V^* we can decode any open formula by an equation $\phi^* = y_{\mathrm{t}}$. If $\Sigma \models \phi$, then $\Sigma \cup \{\neg\phi\}$ is not satisfiable. So the equations of Γ do not have any non-trivial subdirectly irreducible model, and then no non-trivial model at all. Then from Γ we can prove $x = y$ by Birkhoff completeness theorem.

If $\Sigma \not\models \sigma$, then $\Sigma \cup \{\neg\phi\}$ is satisfiable in a model $\mathcal{M}$. If $\mathcal{M}$ is a singleton, by $\mathcal{M} \models \Sigma \cup \{\neg\phi\}$ and (i) we have that $\mathcal{M} \models \phi$ must holds: absurd. If $\mathcal{M}$ is non-singleton, then in $\mathrm{Fa}(\mathcal{M})$ the equations of Γ must hold. Since Γ has a non-trivial model, it can not prove $x = y$. $\square$

8.3.2 Applications to graph theory

We develop the very basics of the theory of graphs with factor algebras. Here the method is more important than the results. We show that it is possible to characterize notions of graph theory by using equations. We believe that building bridges between different fields of mathematics is always fruitful.

In this section the type is fixed, namely $\tau = \{E\}$, where E is a binary relation. First we observe that a τ-factor algebra $\mathbf{A} = \{A, e\}$, where e is a quaternary operation, univocally determines a digraph $G_{\mathbf{A}} = (V, E_{\mathbf{A}})$, where $V = A$ and

$$E_{\mathbf{A}} = \{(a, b) \in A^2 : e(x, y, a, b) = a \text{ for all } a, b \in A\}$$

The digraph G is called the *factor graph* of $\mathbf{A}$.

Conversely, given a digraph $G = (V, E)$ we define the algebra $\mathbf{A}_G$, whose universe is V and whose operation e is defined as:

$$e(x, y, a, b) = \begin{cases} a & \text{if } (x, y) \in E; \\ b & \text{otherwise.} \end{cases}$$

There is a bijection between the class of non-singleton graphs Gr^* and Fa_τ^*. As in the general case, this does not hold for singleton structures.

We use the notation $e([x_0, x_1, \ldots, x_n], y, a, b)$ for the expression

$$e(x_0, x_1, e(x_1, x_2, e(\ldots e(x_{n-1}, x_n, a, b), b) \ldots), b).$$

Notice that $e([x_0, x_1], a, b) \equiv e(x_0, x_1, a, b)$.

Proposition 8.3.3. *Let $G = \{V, E\}$ be a digraph and $\mathbf{A}_G$ the corresponding factor algebra with factor term e. Then we have that:*

$$v_0 \to_E v_1 \to_E \ldots \to_E v_n \quad \textit{iff} \quad \forall ab \; e([v_0, v_1, \ldots, v_n], a, b) = a.$$

Proof. Induction on the number of nodes n. □

Proposition 8.3.4. *Under the hypotheses of proposition 8.3.3, the digraph G is:*

1. *without self-loops* $\Leftrightarrow e(x, x, a, b) = b$
2. *symmetric* $\Leftrightarrow e(x, y, a, b) = e(y, x, a, b)$
3. *asymmetric* $\Leftrightarrow e(x, y, u(y, x, a, b), b) = b.$
4. *transitive* $\Leftrightarrow e(e(x, y, x, y), e(y, z, z, y), a, e(x, y, b, a)) = a.$
5. *acyclic* $\Leftrightarrow$ *for every* $n \geq 0$, $e([x_0, x_1, \ldots, x_n, x_0], a, b) = b.$
6. *bipartite* $\Leftrightarrow$ *for every* $n \geq 0$, $e([x_0, x_1, \ldots, x_{2n}, x_0], a, b) = b.$

7. *an oriented graph* $\Leftrightarrow$ *it is without* ≤ 2*-cycles* $\Leftrightarrow$ $e(x, y, e(y, x, a, b), b) = b$.

8. *a tournament* $\Leftrightarrow$ *it is an oriented graph such that for every* $u, v \in V$*, either* $(u, v) \in E$ *or* $(v, u) \in E$ $\Leftrightarrow$ *it is an oriented graph that satisfies the equation* $e(x, y, e(y, x, a, b), e(y, x, b, a)) = b$.

Proof. All points except (5) and (6) are trivial by definition of $\mathbf{A}_G$.

(5) By proposition 8.3.3, $v_0 \to_E v_1 \to_E \ldots \to_E v_0 \Leftrightarrow e([v_0, v_1, \ldots, v_0], y, z) = z$.

(6) A digraph is bipartite iff it does not contain any odd-length cycle. This holds iff $\forall n \geq 0$ $e([x_0, x_1, \ldots, x_{2n}, x_0], y, z) = z$ holds.

□

Conclusions

Part I

We define two denumerable families of infinite sets, $\mathcal{M}_n$ and $\mathcal{G}_n$ $(n \in \omega)$, whose elements are called *restricted regular mute* and *regular mutes* respectively. Since these terms are built inductively, we give a detailed analysis of their structure and their behavior w.r.t. head reductions. This analysis is used in the proof that regular and restricted regular are mute terms.

Furthermore, we prove that, for each $n \in \omega$, the set $\mathcal{M}_n$ is graph easy. The proof is technical, and relies upon the syntactical structure of restricted hereditarily n-ary terms. The main technique we use is a generalization of forcing for graph models introduced for the first time in [13].

A natural generalization of the result presented in this thesis would be a proof that $\bigcup_{n=1}^{\infty} \mathcal{M}_n$ is a graph easy set. Using an ultraproduct technique developed in [20] we have reduced the general problem to the graph-easiness of $\bigcup_{n\in E} \mathcal{M}_n$ for each finite subset E of natural numbers. Nevertheless, given naturals $n_1 < n_2$, our approach is problematic when dealing with $\mathcal{M}_{n_1}$ and $\mathcal{M}_{n_2}$ simultaneously: the elements $\epsilon_1, \dots, \epsilon_{n_2}$, as defined in the proof of Theorem 4.3.4, force new, unwanted elements to belong to the interpretation of the elements of $\mathcal{M}_{n_1}$. A simpler question concerns the graph-easiness of $\mathcal{M}_1 \cup \mathcal{M}_2$, but it still embodies the difficulty of the general problem.

The other natural question concerns the graph-easiness of the sets $\mathcal{G}_n$ of n-regular mutes, namely:

are the sets $\mathcal{G}_n$ graph-easy?

Here the technique applied to prove graph-easiness of $\mathcal{M}_n$ cannot be directly applied, since it uses the particular form of restricted hereditarily n-ary terms. We were able to prove rather easily a generalization of Lemma 4.3.1 for elements of sets $T_n[\bar{x}]$, but we couldn't overcome Lemma 4.3.2. An attempt of generalization fails at the inductive step, that shows an high technical complexity.

Part II

In this thesis we introduce the notion of factor algebra of an arbitrary first-order type. We develop the basic theory of congruences for this class of algebras by the notion of splitting pair. We also provide necessary and sufficient conditions in order that a factor algebra of first-order type to be simple, subdirectly irreducible or directly indecomposable.

We apply factor algebras to algebraic logic. Given a first-order type τ, we define the class functions Str and Fa, which are bijective functions between the class of non-trivial τ-factor algebras and the class of non-singleton τ-structures. We extend this translation to τ-formulas and equations between terms. This translation has a semantical meaning: given a structure $\mathcal{M}$ and its corresponding factor algebra $\mathbf{A}$, a formula ϕ and its corresponding equation $\phi^* = y_{\mathrm{t}}$, we have that $\mathcal{M} \models \phi$ iff $\mathbf{A} \models \phi^* = y_{\mathrm{t}}$.

We introduce the type τ_{CL} of propositional logic, consisting of denumerable many propositional variables. We apply the developed algebraic techniques to study propositional logic. We prove that the variety $\mathsf{FA}_{\tau_{CL}}$ generated by the τ_{CL}-factor algebras has a simple axiomatization and show that ϕ is a tautology iff the equation $\phi^* = y_{\mathrm{t}}$ holds in $\mathsf{FA}_{\tau_{CL}}$. This implies that axioms defining $\mathsf{FA}_{\tau_{CL}}$ are suitable rules of a calculus in algebraic logic. For such calculus we introduce a term rewriting term system that it is shown confluent and terminating.

Our work is the beginning of a general algebraic logic based on τ-factor algebras. There are various areas in which we think they can produce some good results.

Thanks to the algebraic calculus we have developed for propositional logic, the very first task is to build a purely algebraic propositional logic and provide algebraic proofs of classical propositional theorems such as compactness.

Another immediate application of the theory of τ-factor algebras is the analysis of theories admitting elimination of quantifiers. Thanks to the fact that τ-factor algebras can algebraize all open formulas, it is possible to replace all sentences of such a theory with equations. Such theories are consequently reduced in a purely algebraic setting.

This particular problem dodges the main problem the translation has, i.e., the algebraization of the existential quantifier. The challenging part of this issue is to find a simple way to do this. Many algebraizations of first-order logic had been proposed, but all of them are somehow too complex for applications. If one succeed in this main problem, then the next task is to provide algebraic proofs also of first-order theorems, like Gödel's Completeness Theorem.

We have developed rather naturally our calculus for propositional logic. The fact that the $n+2$-ary function $\hat{R}(\bar{a}, x, y)$ project on the $n+1$ or in the $n+2$ position reflects the excluded middle law. So a last but not least open problem is to provide an algebraic calculus to other logics by using generalizations of τ-factor algebras. In order to do so, one must characterize which are the algebraic structures that semantically interprets the axioms of the logic. The main issue is that the class of these algebras may be too complex. For example, the class of Heyting Algebras, that corresponds to intuitionistic logic, has infinitely many subdirectly irreducible members [23, pag.66]. This may imply that the new τ-factor algebras must be infinitely valued, so that they are not ordinary algebraic structures.

Bibliography

[1] Alessi, F., Dezani-Ciancaglini, M., Honsell, F.: Filter models and easy terms, Italian Conference on Theoretical Computer Science, LNCS 2202, Springer-Verlag, 17–37, 2001.

[2] Alessi, F. , Dezani-Ciancaglini, M., Lusin, S. : Intersection types and domain operators, Theoretical Computer Science, 316(1), 25–47, 2004.

[3] Alessi, F., Lusin, S.: Simple easy terms, in S. van Bakel (ed.), Intersection Types and Related Systems, ENTCS 70, Elsevier, 1–18, 2002.

[4] Baeten, J., Boerboom, B.: Omega can be anything it should not be, in Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Serie A, Indag. Mathematicae 41, 111–120, 1979.

[5] Barendregt, H.P.: Some extensional term models for lambda calculi and combinatory logics, PhD Thesis, Utrecht University, 1971.

[6] Barendregt, H.P.: The lambda-calculus, its syntax and semantics, Studies in Logic vol. 103, North_Holland, revised edition 1984.

[7] Barendregt, H.P.: Representing 'undefined' in lambda calculus. J. Functional Programming 2(3): 367–374, 1992.

[8] Berarducci, A.: Infinite λ-calculus and non-sensible models, in Logic and Algebra, eds. A. Ursini and P. Agliano, Lecture Notes in Pure and Applied Mathematics 180, Marcel Dekker Inc., 339–377, 1996.

[9] Berarducci, A., Intrigila, B.: Some new results on easy λ-terms, Theoretical Computer Science 121, 71–88, 1993.

[10] Berarducci, A., Intrigila, B.:. Church-Rosser lambda-theories, Infinite lambda-terms and consistency problems. In Wilfrid Hodges, Martin Hyland, Charles Steinhorn, and John Truss, editors, Logic: from Foundations to Applications, chapter 2, pages 33-58. 1996.

[11] Berline, C.: From Computation to foundations via functions and application: The lambda-calculus and its webbed models, Theoretical Computer Science 249, 81–161, 2000.

[12] Berline, C.: Graph models of λ-calculus at work, and variations. Mathematical Structures in Computer Science 16, 185–221, 2006.

[13] Berline, C., Salibra, A.: Easiness in graph models, Theoretical Computer Science 354, 4–23, 2006.

[14] Bertini Y.: La notion d'indéfini en lambda-calcul. PhD thesis, Université de Savoie, 2005.

[15] Boyer, C.B: Storia della Matematica, Arnoldo Mondadori editore, 1980.

[16] Bucciarelli, A., Salibra, A.: The minimal graph-model of lambda-calculus, 28th Internat. Symp. on Math. Foundations of Comput. Science, LNCS 2747, Springer-Verlag, 300–307, 2003.

[17] Bucciarelli, A., Salibra, A.: The sensible graph theories of lambda-calculus, Proceedings of the 19th Annual Symposium on Logic in Computer Science (LICS'04), IEEE, 276–285, 2004.

[18] Bucciarelli, A., Salibra, A.: Graph lambda-theories, Mathematical Structures in Computer Science 18(5), 975–1004, 2008.

[19] Bucciarelli, A., Ehrhard, T., Manzonetto, G.: Not Enough Points Is Enough. in CSL 2007, LNCS 4646, 298-312, Springer 2007.

[20] Bucciarelli, A., Carraro, A., Salibra, A.: Minimal lambda-theories by ultraproducts. In D. Kesner and P. Viana: Proceedings Seventh Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2012), EPTCS 113, 61–76, 2012.

[21] Bucciarelli, A., Carraro, A., Favro, G. Salibra, A.: Graph easy sets of lambda terms, accepted paper with minor changes, Journal of Theoretical Computer Science, Elsevier.

[22] Burris, S.: Discriminator varieties and symbolic computation, Journal of Symbolic Computation Volume 13, Issue 2, 175–207, February 1992.

[23] Burris, S., Sankappanavar, H. P.: A Course in Universal Algebra, Millennium Edition. Free online edition, 2013.

[24] Burris, S., Legris, J.: The Algebra of Logic Tradition, The Stanford Encyclopedia of Philosophy (Spring 2015 Edition), Edward N. Zalta (ed.), 2015.

[25] Carraro, A., Salibra, A.: Easy lambda-terms are not always simple, RAIRO - Theor. Inform. and Applic. 46(2), 291–314, 2012.

[26] Coppo, M., Dezani-Ciancaglini, M., Honsell, F., Longo, G.: Extended type structures and filter lambda models, Logic colloquium '82 (Florence, 1982), North-Holland, Amsterdam (1984) 241–262.

[27] Church, A.: The calculi of lambda conversion. Princeton University Press, 1941.

[28] Cvetko-Vah, K., Salibra, A.: The connection of skew Boolean algebras and discriminator varieties to Church algebras Algebra Universalis, vol. 73(3), pp. 369-390, 2015.

[29] Nachum Dershowitz, N., and Jean-Pierre Jouannaud, J.P.: Rewrite Systems, Chapter 6 of Handbook of Theoretical Computer Science Volume B: Formal Methods and Semantics, ed. J. van Leeuwen, pages 243 320, North-Holland, Amsterdam, 1990.

[30] Enderton, H. A Mathematical Introduction to Logic, Harcourt/Academic Press. 2001.

[31] Engeler, E.: Algebras and combinators, Alg. Univ. 13(3), 289–371, 1981.

[32] Galatos, N., Jipsen, P., Kowalski, T., Ono, H.: Residuated Lattices: An Algebraic Glimpse at Substructural Logics, Elsevier, 2007.

[33] Givant, S.R., Tarski, A.: A formalization of Set Theory without variables, American Mathematical Society, 1987.

[34] Halmos, P.: Algebraic Logic, Chelsea Publishing, New York, 1962.

[35] Hasegawa, M.: Recursion from cyclic sharing: Traced monoidal categories and models of cyclic lambda calculi. In P. de Groote (ed.), Proceedings of Typed Lambda Calculi and Applications (TLCA'97), LNCS 1210, 196–213, 1997.

[36] Henkin, L., Monk, J.D., Tarski, A.: Cylindric Algebras: Part I, in Studies in logic and the foundation of mathematics, North-Holland, 1975.

[37] Herstein, I.N.: Algebra. Editori Riuniti, 1982.

[38] Intrigila B.: A problem on easy terms in lambda calculus. Fund. Inform, 15(1), 99–106, 1991.

[39] Jacopini, C.: A condition for identifying two elements of whatever model of combinatory logic, in C. Böhm, ed., LNCS 37, Springer Verlag, 213–219, 1975.

[40] Jacopini, C., Venturini-Zilli, M.: Easy terms in the lambda-calculus, Fundamenta Informaticae VIII.2, 225–233, 1985.

[41] Jiang, Y.: Consistency of a λ-theory with n-tuples and easy terms, Archives of Math. Logic, 34(2), 79–96, 1995.

[42] Kamin, S., Lévy, J.J.: Two generalizations of the recursive path ordering, unpublished note, department of Computer Science, University of Illinois, 1980.

[43] Kerth, R.: Isomorphism and equational equivalence of continuous λ-models, Studia Logica 61, 403–415, 1998.

[44] Kerth, R.: Isomorphisme et équivalence équationnelle entre modèles du λ-calcul, Thèse, Université Paris 7, 1995.

[45] Kerth, R.: Forcing in stable models of untyped λ-calculus, Indagationas Mathematicae 10 , 59–71, 1999.

[46] Kleene, S.C.: λ-definability and recursiveness, Duke Math. J. 2, 340–353, 1936.

[47] Kleene, S.C.: General recursive functions of natural numbers, Math. Ann. 112, 727–742, 1936.

[48] Krivine, J. L.: Lambda-Calculus, Types and Models. Masson, Paris, and Ellis Horwood, Hemel Hempstead, 1993.

[49] Longo, G.: Set-theoretical models of λ-calculus : theories, expansions and isomorphisms, Annals of Pure and Applied Logic 24, 153–188, 1983.

[50] Manca, V., Salibra, A.: First-order theories as many-sorted algebras. Notre Dame Journal of Formal Logic Volume 25, Number 1, 86-94, 1994.

[51] McKenzie, R.N., McNulty, G.F., Taylor, W.F.: Algebras, Lattices, Varieties, Volume I, Wadsworth Brooks, 1987.

[52] Meinke, K., Tucker, J.V.: Universal algebra. Journal of Symbolic Logic. Oxford University Press 1–189 (1992),

[53] Mycielski, J.: Equational treatment of first-order logic, Algebra Universalis, Volume 33, Issue 1, March 1995.

[54] Newman, M.H.A. : On theories with a combinatorial definition of “equivalence”. Annals of Mathematics, 43(2):223–243, 1942.

[55] Ohlebusch, E.: A Note on Simple Termination of Infinite Term Rewriting Systems, Technical report, 1992.

[56] Plotkin, G.: A set-theoretical definition of application, Memorandum MIP-R-95, School of artificial intelligence, University of Edinburgh, 1972.

[57] Salibra, A., A. Ledda, A., Paoli, A., Kowalski, T.: Boolean-like algebras, Algebra Universalis, Vol. 69(2), 113-138, 2013.

[58] Salibra, A., Ledda, A., Paoli, F., : Factor Varieties, Soft Computing, 2015.

[59] Severi, P.G., de Vries, F.J.: Weakening the Axiom of Overlap in the Infinitary Lambda Calculus. In RTA, LIPICS vol. 10, 313–328, 2011.

[60] Severi, P.G., de Vries, F.J.: Decomposition and cardinality of intervals in the lattice of meaningless sets, in WOLLIC, LNAI 6642, Springer, 210–227, 2011.

[61] Statman, R.: Two recursion theoretic problems in lambda calculus. Manuscript, Department of Mathematics, Carnegie Mellon University, Pittsburgh, USA, 1987.

[62] Statman, R.: The Visser Fixed Point Theorem. Manuscript, Department of Mathematics, Carnegie Mellon University, Pittsburgh, USA, 1990.

[63] Tarski, A., Thompson, F.B.: Some general properties of cylindric algebras. Bulletin of the American Mathematical Society. Vol. 58. No. 1, 1952.

[64] Terese: Term Rewriting Systems, Cambridge Tracts in Theoretical Computer Science, Vol. 55, Cambridge University Press, 2003.

[65] Turing, A.: Computability and λ-definability. J. Symbolic Logic 2(4), 153–163, 1937.

[66] Turing, A.: The þ-function in λ-K-conversion. J. Symbolic Logic 2(4), 164, 1937.

[67] Vaggione D., Equational characterization of the quaternary discriminator, Algebra Universalis, 43, pp.99-100, 2000.

[68] Visser, A.: Numerations, λ-calculus and arithmetic, To H.B. Curry: Essays on Combinatory Logic, Lambda-Calculus and Formalism (J.R. Hindley and J.P. Seldin eds.), Academic Press, New York, p. 259-284, 1980.

[69] Zylberajch, C.: Syntaxe et sémantique de la facilité en λ-calcul, Thèse, Université Paris 7, 1991.